Bölüm 6: Çevrim İçi Platformlar

Algoritmik Oyun Kuramı

İki taraflı pazarlar | Ağ etkileri ve kritik kitle | Platform fiyatlandırması | Sponsorlu arama: GSP ve VCG | İtibar sistemleri

Sercan KÜLCÜ · Giresun Üniversitesi · Bilgisayar Mühendisliği

© Sercan Külcü. Tüm hakları saklıdır. · Sunumun etkileşimli HTML sürümü: https://sercankulcu.github.io/files/agt/slides/Bolum_06_Cevrim_Ici_Platformlar.html

PLATFORMLAR VE PAZARLAR

Oyunla Başlangıç: Sponsorlu Arama Açık Artırması

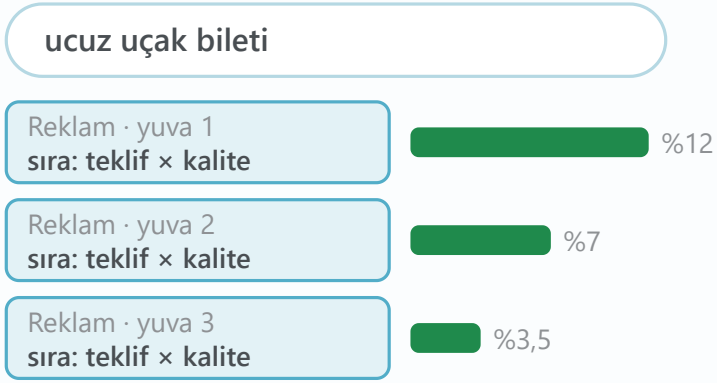
Oyunun kuralı

Reklamverenler tık başına teklif verir; reklamlar teklif × kalite puanına göre üç yuvaya sıralanır ve üstteki yuva daha çok tıklanır.

Merak sorusu

- Bir tık size 8 TL kazandırıyorsa 8 TL teklif etmek en iyisi mi?
- En yüksek teklifi veren her zaman en üst yuvayı mı alır?

[Sponsorlu Arama Açık Artırması oyunu →]



Çubuk: yuvanın tıklanma oranı (Search Ads senaryosu)

PLATFORMLAR VE PAZARLAR

Öğrenme Çıktıları

Platformu modellemek

İki taraflı pazarı, çapraz ağ etkilerini ve platformu bir mekanizma tasarımcısı olarak tanımlamak.

Kritik kitleyi hesaplamak

Basit bir benimseme modelinde dengeleri, eşiği ve “devrilme” (tipping) dinamiğini bulmak.

Fiyat yapısını açıklamak

İki taraflı sayısal örnekte hangi tarafın neden sübvanse edildiğini göstermek.

Reklam açık artırmaları

GSP ve VCG ödemelerini aynı tekliflerle hesaplamak; GSP'de dürüst teklifin neden denge olmadığını göstermek.

İtibarı ölçmek

Bayesçi ortalamayla sıralamak; sahte değerlendirmenin ne zaman kârlı olduğunu hesaplamak.

Rekabeti tartışmak

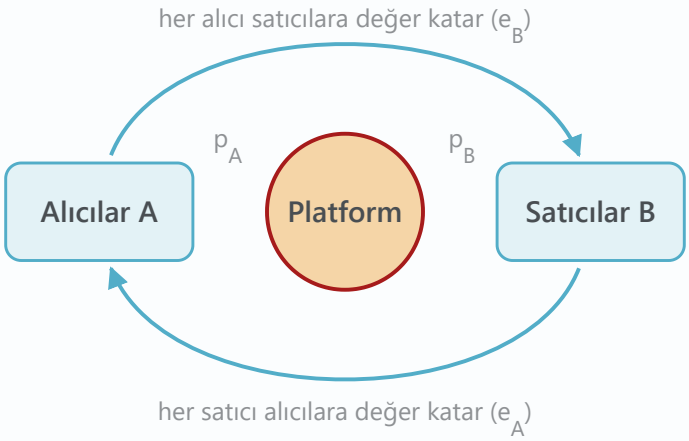
Devrilme, geçiş maliyeti, çoklu kullanım ve düzenlemenin (AB Dijital Piyasalar Yasası, DMA) platform rekabetine etkisini yorumlamak.

Çevrim İçi Platform Nedir?

- Çevrim içi platform: kullanıcıları, satıcıları ve hizmet sağlayıcıları buluşturan dijital ekosistem (aracı).
- Değer, platformun kendisinden çok tarafların **birbirine erişiminden** doğar: iki taraflı pazar (two-sided market).
- Platform kuralları koyar: kim katılır, kim kimi görür, kim ne öder → bir **mekanizma**.

Tür	1. taraf	2. taraf	Örnek
E-ticaret pazar yeri	alıcılar	satıcılar	Trendyol, Hepsiburada, Amazon
Paylaşım ekonomisi	misafir / yolcu	ev sahibi / sürücü	Airbnb, BlaBlaCar, Uber
Sosyal medya	kullanıcılar	reklamveren, içerik üreticisi	Instagram, TikTok, Facebook
Arama motoru	arayanlar	reklamverenler	Google, Bing

İki Taraflı Pazarlar ve Çapraz Ağ Etkisi



Çapraz ağ etkisi

Bir taraftaki katılımcı sayısı, diğer tarafın elde ettiği değeri belirler (çok satıcı → alıcı için çok seçenek).

Fiyat yapısı

Platform yalnız toplam fiyatı değil, ücretin taraflar arasında nasıl bölüşüleceğini de seçer (p_A, p_B).

Tavuk-yumurta sorunu

Satıcı yoksa alıcı gelmez, alıcı yoksa satıcı gelmez: başlangıçta bir taraf “çekilmelidir”.

PLATFORMLAR VE PAZARLAR

Platform Bir Mekanizma, Kullanıcılar Oyuncu

- Platform tasarımı (ücret, sıralama, puanlama kuralları) kullanıcıların stratejik davranışlarını yönlendirir.
- Kullanıcılar teşviklere göre strateji seçer; ortaya çıkan davranış kurallar altındaki **Nash dengesidir**.

Platformun teşvikleri

+ Bonus: yoğun bölgede sürücüye ek ödeme (Uber)
+ İndirim, kupon, ücretsiz deneme
+ Sıralama ve öneri algoritması → görünürlük
+ Puan ve rozetler (güvenilir satıcı)

Kullanıcıların stratejik tepkileri

– Sahte yorumlar ve puan şişirme
– Çoklu hesap açma, bonus istismarı
– Algoritmayı manipüle etme (tıklama çiftlikleri)
– Misilleme amaçlı kötü değerlendirme

Doğru teşvik → işbirliği ve verimlilik. Yanlış teşvik → manipülasyon ve güven kaybı.

AĞ ETKİLERİ

Ağ Etkisi Türleri

Doğrudan (aynı taraf)

WhatsApp: ne kadar çok arkadaşın kullanıyorsa uygulama senin için o kadar değerli.

Dolaylı / çapraz

Uber: çok sürücü → kısa bekleme → çok yolcu → sürücüye daha çok iş.

Negatif

Kalabalık → tıkanıklık, bilgi kirliliği; aynı taraftaki satıcılar birbirine rakiptir.

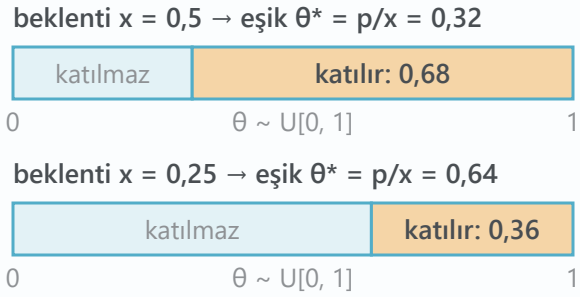
Yerel

Değer tüm kullanıcılardan değil yakın çevreden gelir (şehir, arkadaş grubu); rekabet yerelde kazanılır.

Pozitif ağ etkisi bir **geri besleme döngüsü** kurar: büyüyen daha da büyür, küçülen daha da küçülür. Pazar bir yöne “devrilebilir”.

Basit Bir Benimseme (Adoption) Modeli

- N = 1000 potansiyel kullanıcı; kullanıcı i'nin tipi $\theta_i \sim U[0, 1]$: platforma ne kadar önem verdiği.
- Nüfusun x oranı katılırsa i'nin faydası $\theta_i \cdot x - p$; burada p = 0,16 (ücret veya katılma çabası).
- Kullanıcı katılır $\Leftrightarrow \theta_i \cdot x \geq p \Leftrightarrow \theta_i \geq p/x$. θ, 0 ile 1 arasında eşit olasılıkla dağıldığından θ'sı p/x'ten büyük olanların oranı $1 - p/x$'tir.

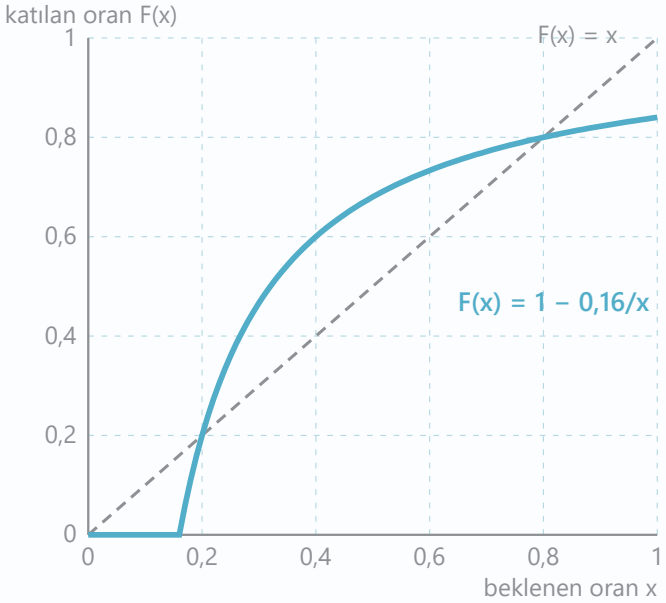


x kişi bekleniyorsa katılan oran: $F(x) = 1 - p/x$ $(x \leq p$ ise 0)

- **Denge**: kendini doğrulayan beklenti, $F(x) = x$.
- **Dinamik**: bu dönem katılanlar gelecek dönemin beklentisini belirler, $x_{t+1} = F(x_t)$.

Easley ve Kleinberg (2010, Bölüm 17) ağ etkisi modelinin basitleştirilmiş biçimi.

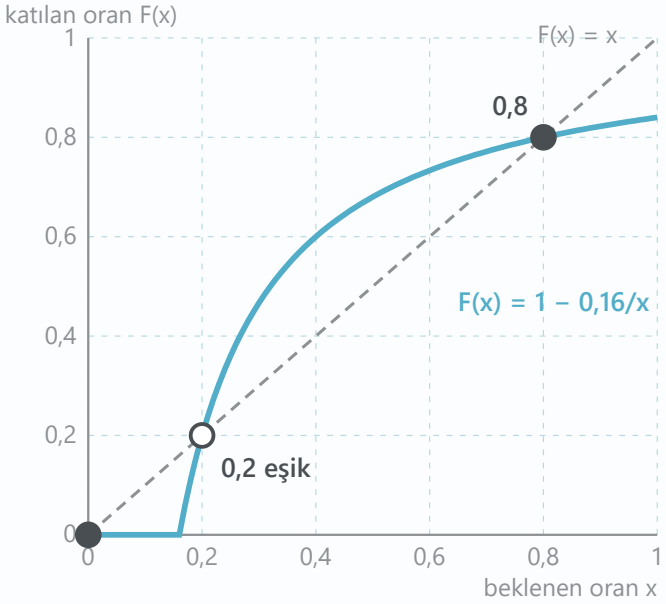
Adım Adım: Kritik Kitle ve Devrilme



t	$x_0 = 0{,}25$	$x_0 = 0{,}19$
0		
1		
2		
3		
4		
5		
6		

Eğri F(x) = 1 − 0,16/x: “x oranı katılacak” beklentisiyle gerçekten katılanların oranı. x ≤ 0,16 iken kimse katılmaz.

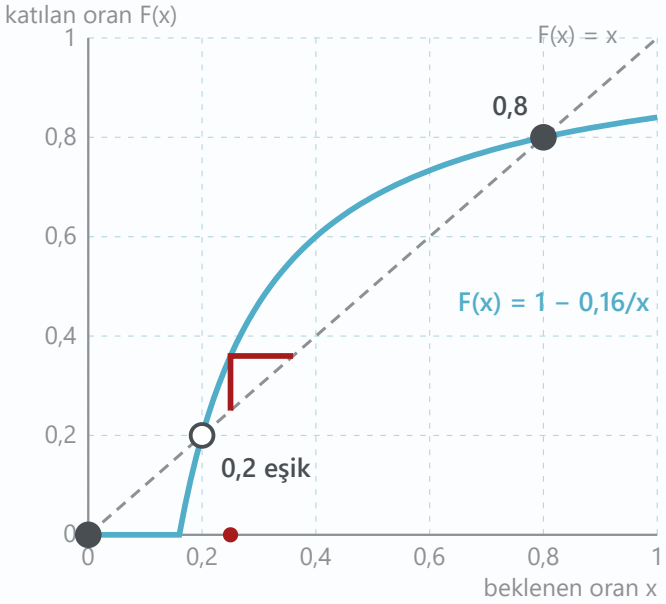
Adım Adım: Kritik Kitle ve Devrilme



t	x_0 = 0,25	x_0 = 0,19
0		
1		
2		
3		
4		
5		
6		

Dengeler F(x) = x: 1 − 0,16/x = x, x ile çarpılınca $x^2 - x + 0{,}16 = 0$ → **x = 0,2 ve x = 0,8**; ayrıca x = 0 (kimse yok). İçi boş nokta kararsız denge: küçük bir sapma dinamiği dengeden uzaklaştırır.

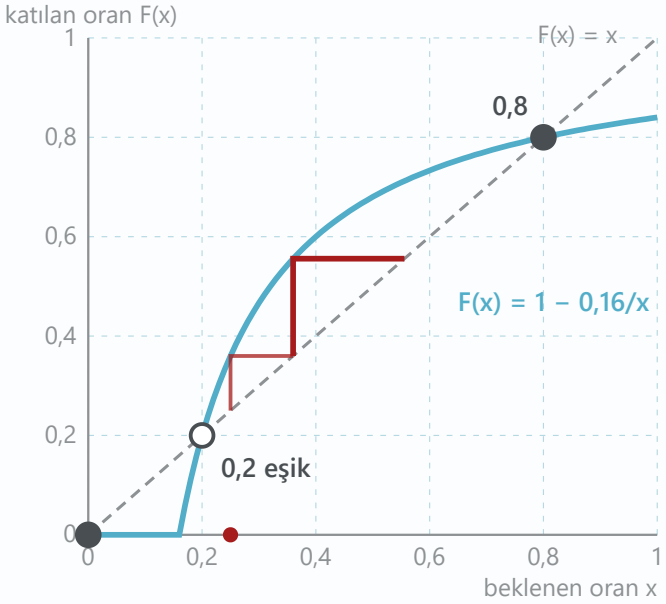
Adım Adım: Kritik Kitle ve Devrilme



t	x_0 = 0,25	x_0 = 0,19
0	0,25	
1	0,36	
2		
3		
4		
5		
6		

250 kullanıcı bekleniyor (x_0 = 0,25): $\theta \geq 0{,}16/0{,}25 = 0{,}64$ olanlar katılır → x_1 = **0,36**.

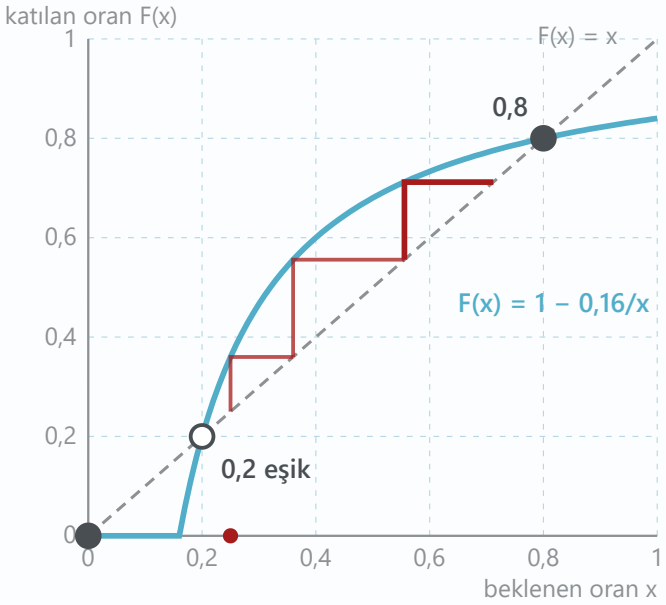
Adım Adım: Kritik Kitle ve Devrilme



t	x_0 = 0,25	x_0 = 0,19
0	0,25	
1	0,36	
2	0,556	
3		
4		
5		
6		

Katılım beklentiyi aştı; yeni beklenti 0,36 ile θ ≥ 0,444 olanlar katılır → x_2 = **0,556**.

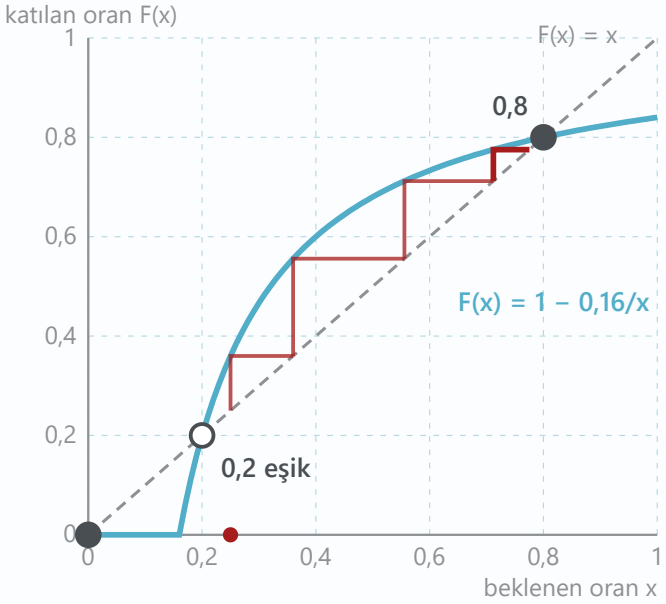
Adım Adım: Kritik Kitle ve Devrilme



t	x_0 = 0,25	x_0 = 0,19
0	0,25	
1	0,36	
2	0,556	
3	0,712	
4		
5		
6		

Geri besleme sürüyor: x_3 = 1 − 0,16/0,556 = **0,712**.

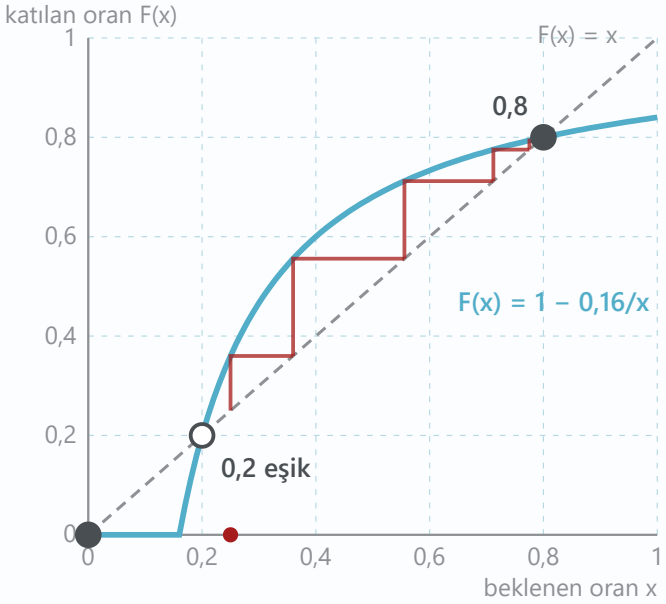
Adım Adım: Kritik Kitle ve Devrilme



t	x_0 = 0,25	x_0 = 0,19
0	0,25	
1	0,36	
2	0,556	
3	0,712	
4	0,775	
5		
6		

x_4 = **0,775**; adımlar küçülüyor, yol kararlı dengeye yaklaşıyor.

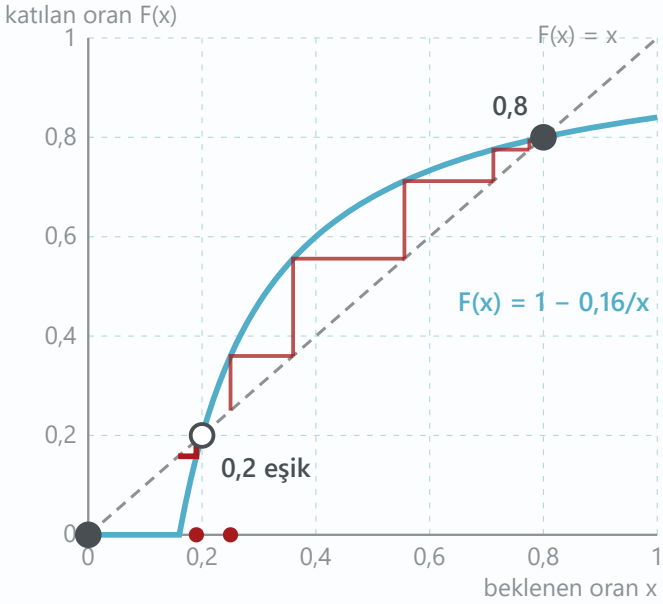
Adım Adım: Kritik Kitle ve Devrilme



t	x_0 = 0,25	x_0 = 0,19
0	0,25	
1	0,36	
2	0,556	
3	0,712	
4	0,775	
5	0,794	
6	0,798	

$x_6 \approx 0{,}798 \rightarrow 0{,}8$: pazar platform lehine **devrildi** (800 kullanıcı).

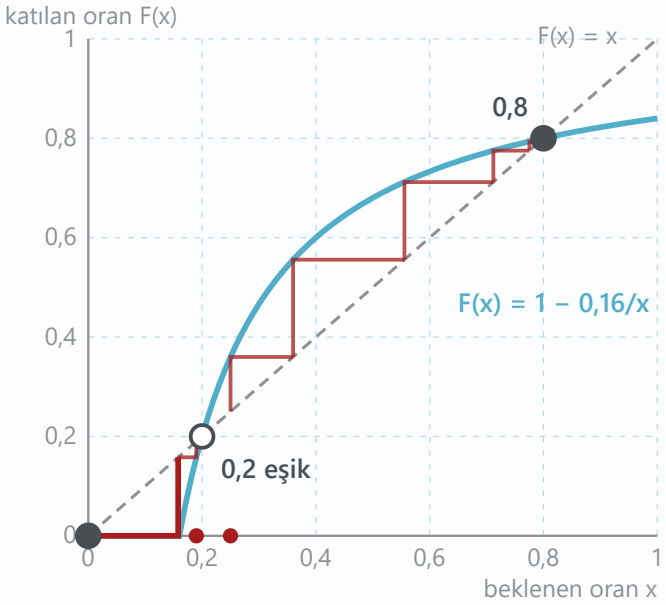
Adım Adım: Kritik Kitle ve Devrilme



t	x_0 = 0,25	x_0 = 0,19
0	0,25	0,19
1	0,36	0,158
2	0,556	
3	0,712	
4	0,775	
5	0,794	
6	0,798	

Eşiğin hemen altından başla: $x_0 = 0{,}19 \rightarrow x_1 = 1 - 0{,}16/0{,}19 =$ **0,158**. Katılım beklentinin altında kaldı.

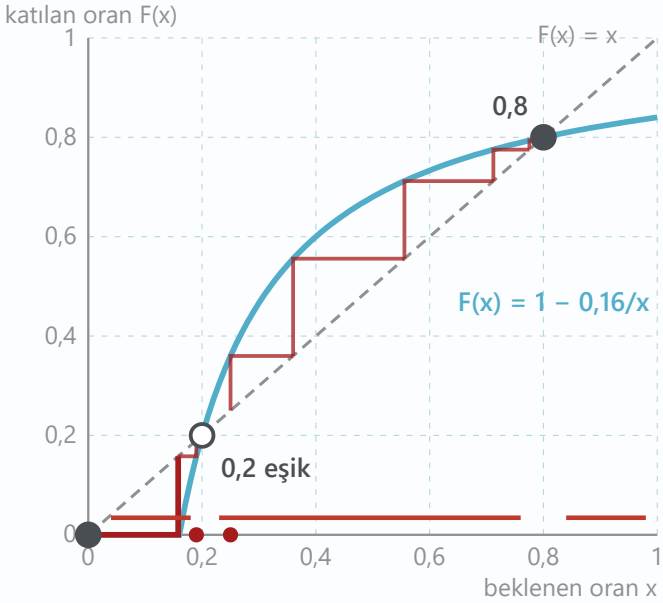
Adım Adım: Kritik Kitle ve Devrilme



t	x_0 = 0,25	x_0 = 0,19
0	0,25	0,19
1	0,36	0,158
2	0,556	0
3	0,712	
4	0,775	
5	0,794	
6	0,798	

$x_1 = 0{,}158 \leq p = 0{,}16 \rightarrow x_2 = \mathbf{0}$: kimse katılmaz, platform çöker.

Adım Adım: Kritik Kitle ve Devrilme



t	x_0 = 0,25	x_0 = 0,19
0	0,25	0,19
1	0,36	0,158
2	0,556	0
3	0,712	
4	0,775	
5	0,794	
6	0,798	

Sonuç: 0 ve 0,8 kararlı, 0,2 kararsız. **Kritik kitle 0,2 (200 kullanıcı)**: beklenti bunun üstündeyse 0,8'e, altındaysa 0'a gidilir.

Kritik Kitleye Ulaşmak: Tasarım Dersleri

p	dengeler	eşik
0,09	0, 0,1, 0,9	0,1
0,16	0, 0,2, 0,8	0,2
0,21	0, 0,3, 0,7	0,3
0,25	0 ve 0,5 (teğet)	0,5
0,3	yalnız 0	—

Ücret düştükçe eşik düşer. $x^2 - x + p = 0$ denkleminin gerçek kökü olması için diskriminant $1 - 4p \geq 0$, yani $p \leq 0{,}25$ olmalı; $p > 0{,}25$ olursa pazar hiç oluşmaz.

Başlangıç sübvansiyonu

Ücretsiz dönem, bonus: p küçülür, kritik kitle düşer.

Tohumlama (seeding)

İlk içeriği, satıcıyı veya sürücüyü platform kendisi sağlar.

Beklenti yönetimi

"Herkes burada" algısı x_0'ı eşiğin üstüne taşır.

Tek başına değer

Ağ olmadan da işe yarayan bir araçla başla, ağı sonra kur.

Örnek: WhatsApp kullanıcı arttıkça değeri katlanarak büyüdü; Friendster ise kullanıcı kaybı başlayınca aynı döngüyle ters yönde çöktü.

PLATFORM FİYATLANDIRMASI

İki Taraflı Fiyatlandırma İlkeleri

- Platform toplam fiyat **düzeyinin** yanında fiyat **yapısına** da karar verir: ücreti hangi taraf, ne kadar öder? (Rochet ve Tirole, 2003)
- Diğer tarafa daha çok değer katan veya fiyata daha duyarlı olan taraf ucuzlatılır, hatta **sübvanse edilir** (negatif fiyat).
- Kayıp, sübvanse edilen tarafın çektiği diğer taraftan alınan ücretle telafi edilir.

Arama / sosyal medya
Kullanıcı ücretsiz, reklamveren öder.

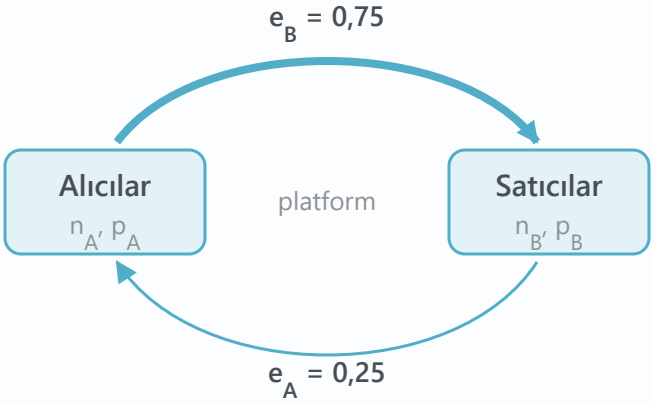
Kredi kartı
Kart sahibine puan, işyerinden komisyon.

Pazar yeri
Alıcıya kupon, satıcıdan komisyon.

Gelir modelleri
Komisyon, üyelik, reklam, işlem başına ücret.

Basit doğrusal modelde (Armstrong, 2006) kural: **p_A = kâr payı − bir A katılımcısının B tarafına kattığı değer**. İkinci terim büyükse fiyat negatife düşer. Kâr payı: ağ etkisi olmasaydı platformun bir katılımcıdan alacağı fiyat ile maliyet farkı.

Sayısal Örnek: Pazar Yeri Modeli



- n: katılımcı (bin kişi), p: aylık ücret (TL). Bir katılımcıya hizmet vermenin platforma ek maliyeti yok (marjinal maliyet 0).
- Satıcı için alıcı çok değerli ($e_B = 0{,}75$), alıcı için satıcı çeşitliliği daha az ($e_A = 0{,}25$).

$$n_A = 20 + 0{,}25 \cdot n_B - p_A$$
$$n_B = 80 + 0{,}75 \cdot n_A - p_B$$
$$\Pi = p_A \cdot n_A + p_B \cdot n_B$$

Soru: kârı maksimize eden (p_A, p_B) nedir? Hangi taraf sübvanse edilir?

Adım Adım: Hangi Taraf Sübvanse Edilir?

1. Katılımları karar değişkeni yap: $p_A = 20 + 0{,}25\, n_B - n_A$, $p_B = 80 + 0{,}75\, n_A - n_B$

Senaryo	e_A / e_B	p_A	p_B	n_A	n_B	Π

Fiyat yerine katılımları seçmek hesabı kolaylaştırır: her (n_A, n_B) çiftine tek bir fiyat çifti karşılık gelir.

Betikte (p_A, p_B) ızgarası 0,5 TL adımla tarandı: en yüksek kâr (−5, 50) noktasında.

Adım Adım: Hangi Taraf Sübvanse Edilir?

1. Katılımları karar değişkeni yap: $p_A = 20 + 0{,}25\, n_B - n_A$, $p_B = 80 + 0{,}75\, n_A - n_B$

2. $\Pi = 20\, n_A + 80\, n_B + (0{,}25 + 0{,}75)\, n_A n_B - n_A^2 - n_B^2$

Senaryo	e_A / e_B	p_A	p_B	n_A	n_B	Π

Kâr ifadesine yerleştir. $n_A n_B$ terimi iki tarafın birbirine kattığı değeri platformun toplayabileceğini gösterir.

Betikte (p_A, p_B) ızgarası 0,5 TL adımla tarandı: en yüksek kâr (−5, 50) noktasında.

Adım Adım: Hangi Taraf Sübvanse Edilir?

1. Katılımları karar değişkeni yap: $p_A = 20 + 0{,}25\, n_B - n_A$, $p_B = 80 + 0{,}75\, n_A - n_B$

2. $\Pi = 20\, n_A + 80\, n_B + (0{,}25 + 0{,}75)\, n_A n_B - n_A^2 - n_B^2$

3. $\partial\Pi/\partial n_A = 20 + n_B - 2n_A = 0$, $\partial\Pi/\partial n_B = 80 + n_A - 2n_B = 0$

Senaryo	e_A / e_B	p_A	p_B	n_A	n_B	Π

n_B sabit tutulup Π'nin n_A'ya göre türevi alınır (kısmi türev, ∂ ile gösterilir); en iyi noktada iki türev de 0'dır: iki doğrusal denklem.

Betikte (p_A, p_B) ızgarası 0,5 TL adımla tarandı: en yüksek kâr (−5, 50) noktasında.

Adım Adım: Hangi Taraf Sübvanse Edilir?

1. Katılımları karar değişkeni yap: $p_A = 20 + 0{,}25\, n_B - n_A$, $p_B = 80 + 0{,}75\, n_A - n_B$

2. $\Pi = 20\, n_A + 80\, n_B + (0{,}25 + 0{,}75)\, n_A n_B - n_A^2 - n_B^2$

3. $\partial\Pi/\partial n_A = 20 + n_B - 2n_A = 0$, $\partial\Pi/\partial n_B = 80 + n_A - 2n_B = 0$

4. $n_A = 40$, $n_B = 60 \rightarrow p_A = 20 + 15 - 40 = \mathbf{-5}$, $p_B = 80 + 30 - 60 = \mathbf{50}$

Senaryo	e_A / e_B	p_A	p_B	n_A	n_B	Π
Optimum	0,25 / 0,75	−5	50	40	60	**2800**

Çözüm: alıcılara **5 TL ödenir** ($p_A = -5$), satıcılar 50 TL öder. Kâr $\Pi = -200 + 3000 = \mathbf{2800}$.

Betikte (p_A, p_B) ızgarası 0,5 TL adımla tarandı: en yüksek kâr (−5, 50) noktasında.

Adım Adım: Hangi Taraf Sübvanse Edilir?

1. Katılımları karar değişkeni yap: $p_A = 20 + 0{,}25\, n_B - n_A$, $p_B = 80 + 0{,}75\, n_A - n_B$
2. $\Pi = 20\, n_A + 80\, n_B + (0{,}25 + 0{,}75)\, n_A n_B - n_A^2 - n_B^2$
3. $\partial\Pi/\partial n_A = 20 + n_B - 2n_A = 0$, $\partial\Pi/\partial n_B = 80 + n_A - 2n_B = 0$
4. $n_A = 40$, $n_B = 60 \rightarrow p_A = 20 + 15 - 40 = \mathbf{-5}$, $p_B = 80 + 30 - 60 = \mathbf{50}$

Senaryo	e_A / e_B	p_A	p_B	n_A	n_B	Π
Optimum	0,25 / 0,75	−5	50	40	60	**2800**
Alıcıya ücretsiz	0,25 / 0,75	0	47,5	34,6	58,5	**2776,9**

Karşılaştır: alıcı ücretsiz ($p_A = 0$) olursa en iyi $p_B \approx 47{,}5$; daha az alıcı, kâr 2776,9 < 2800.

Betikte (p_A, p_B) ızgarası 0,5 TL adımla tarandı: en yüksek kâr (−5, 50) noktasında.

Adım Adım: Hangi Taraf Sübvanse Edilir?

1. Katılımları karar değişkeni yap: $p_A = 20 + 0{,}25\, n_B - n_A$, $p_B = 80 + 0{,}75\, n_A - n_B$
2. $\Pi = 20\, n_A + 80\, n_B + (0{,}25 + 0{,}75)\, n_A n_B - n_A^2 - n_B^2$
3. $\partial\Pi/\partial n_A = 20 + n_B - 2n_A = 0$, $\partial\Pi/\partial n_B = 80 + n_A - 2n_B = 0$
4. $n_A = 40$, $n_B = 60 \rightarrow p_A = 20 + 15 - 40 = \mathbf{-5}$, $p_B = 80 + 30 - 60 = \mathbf{50}$

Senaryo	e_A / e_B	p_A	p_B	n_A	n_B	Π
Optimum	0,25 / 0,75	−5	50	40	60	**2800**
Alıcıya ücretsiz	0,25 / 0,75	0	47,5	34,6	58,5	**2776,9**
Alıcıdan 10 TL	0,25 / 0,75	10	42,5	23,8	55,4	**2592,3**

Alıcıdan 10 TL alınırsa alıcı sayısı 23,8 bine düşer, satıcılar da azalır: kâr 2592,3.

Betikte (p_A, p_B) ızgarası 0,5 TL adımla tarandı: en yüksek kâr (−5, 50) noktasında.

Adım Adım: Hangi Taraf Sübvanse Edilir?

1. Katılımları karar değişkeni yap: $p_A = 20 + 0{,}25\, n_B - n_A$, $p_B = 80 + 0{,}75\, n_A - n_B$

2. $\Pi = 20\, n_A + 80\, n_B + (0{,}25 + 0{,}75)\, n_A n_B - n_A^2 - n_B^2$

3. $\partial\Pi/\partial n_A = 20 + n_B - 2n_A = 0$, $\partial\Pi/\partial n_B = 80 + n_A - 2n_B = 0$

4. $n_A = 40$, $n_B = 60 \rightarrow p_A = 20 + 15 - 40 = \mathbf{-5}$, $p_B = 80 + 30 - 60 = \mathbf{50}$

5. Kural: $p_A = n_A - e_B \cdot n_B = 40 - 0{,}75 \cdot 60 = -5$ (kâr payı n_A – alıcının satıcılara kattığı değer)

Senaryo	e_A / e_B	p_A	p_B	n_A	n_B	Π
Optimum	0,25 / 0,75	−5	50	40	60	**2800**
Alıcıya ücretsiz	0,25 / 0,75	0	47,5	34,6	58,5	**2776,9**
Alıcıdan 10 TL	0,25 / 0,75	10	42,5	23,8	55,4	**2592,3**

Neden? Fazladan bir alıcı satıcılara $0{,}75 \cdot 60 = 45$ TL değer katar ve platform bunu p_B ile geri alır. $45 > 40$ (kâr payı) → sübvansiyon.

Betikte (p_A, p_B) ızgarası 0,5 TL adımla tarandı: en yüksek kâr (−5, 50) noktasında.

Adım Adım: Hangi Taraf Sübvanse Edilir?

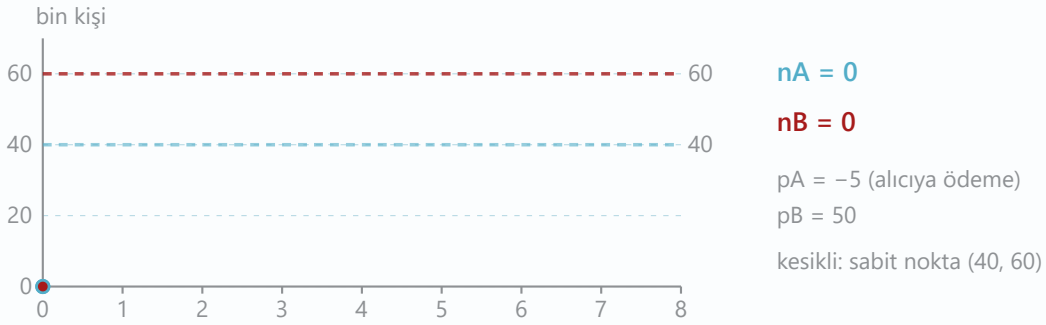
1. Katılımları karar değişkeni yap: $p_A = 20 + 0{,}25\, n_B - n_A$, $p_B = 80 + 0{,}75\, n_A - n_B$
2. $\Pi = 20\, n_A + 80\, n_B + (0{,}25 + 0{,}75)\, n_A n_B - n_A^2 - n_B^2$
3. $\partial\Pi/\partial n_A = 20 + n_B - 2n_A = 0$, $\partial\Pi/\partial n_B = 80 + n_A - 2n_B = 0$
4. $n_A = 40$, $n_B = 60 \rightarrow p_A = 20 + 15 - 40 = \mathbf{-5}$, $p_B = 80 + 30 - 60 = \mathbf{50}$
5. Kural: $p_A = n_A - e_B \cdot n_B = 40 - 0{,}75 \cdot 60 = -5$ (kâr payı n_A − alıcının satıcılara kattığı değer)

Senaryo	e_A / e_B	p_A	p_B	n_A	n_B	Π
Optimum	0,25 / 0,75	−5	50	40	60	**2800**
Alıcıya ücretsiz	0,25 / 0,75	0	47,5	34,6	58,5	**2776,9**
Alıcıdan 10 TL	0,25 / 0,75	10	42,5	23,8	55,4	**2592,3**
Etkiler ters	0,75 / 0,25	25	30	40	60	**2800**

Karşı olgu: çapraz etkiler ters olsaydı (alıcılar satıcılara az değer katsaydı) $p_A = 25$, $p_B = 30$ olurdu; sübvansiyon kalkar.

Betikte (p_A, p_B) ızgarası 0,5 TL adımla tarandı: en yüksek kâr (−5, 50) noktasında.

Kod İzleme: `katilim(pA, pB)`



Çağrı: optimum fiyatlar pA = −5, pB = 50. Katılımı “elle” çözmek yerine taraflar birbirine tepki versin.

Adım 1 / 14

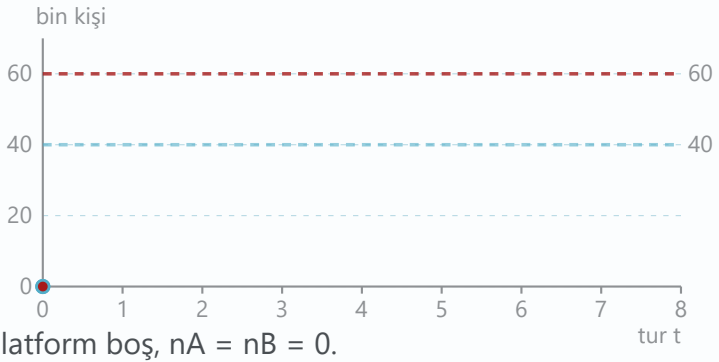
```
▶ def katilim(pA, pB, tur=8):
2   nA, nB = 0, 0
3   for t in range(tur):
4     nA = 20 + 0.25 * nB - pA    # alıcılar
5     nB = 80 + 0.75 * nA - pB    # satıcılar
6   return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `tur = 8`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



nA = 0

nB = 0

pA = −5 (alıcıya ödeme)

pB = 50

kesikli: sabit nokta (40, 60)

Başlangıç: platform boş, nA = nB = 0.

Adım 2 / 14

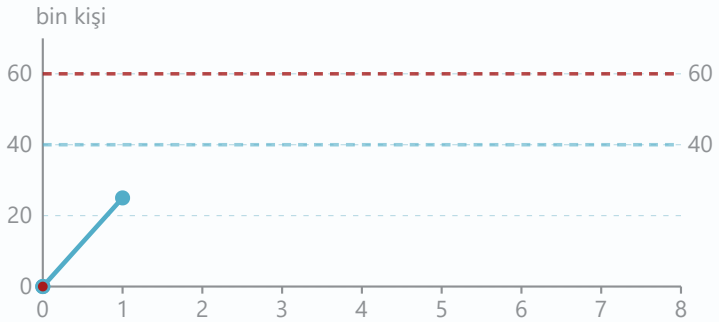
```
1 def katilim(pA, pB, tur=8):
▶   nA, nB = 0, 0
3   for t in range(tur):
4     nA = 20 + 0.25 * nB - pA    # alıcılar
5     nB = 80 + 0.75 * nA - pB    # satıcılar
6   return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `nA = 0` `nB = 0`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



nA = 25

nB = 0

pA = −5 (alıcıya ödeme)
pB = 50

kesikli: sabit nokta (40, 60)

Tur 1: alıcılar mevcut satıcı sayısına tepki verir: nA = 20 + 0,25·0 + 5 = **25**.

Adım 3 / 14

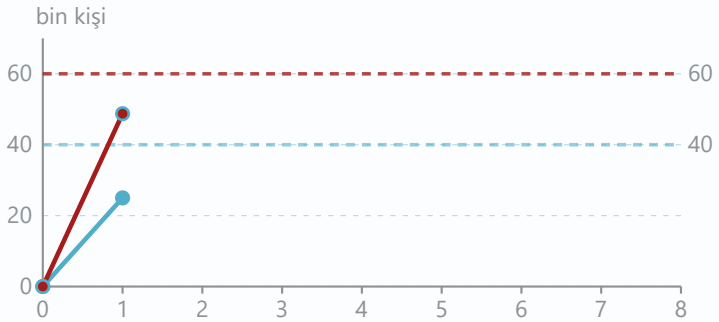
```
1 def katilim(pA, pB, tur=8):
2   nA, nB = 0, 0
3   for t in range(tur):
▶     nA = 20 + 0.25 * nB - pA    # alıcılar
5     nB = 80 + 0.75 * nA - pB    # satıcılar
6   return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `t = 0` `nA = 25.0` `nB = 0.0`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



nA = 25

nB = 48,8

pA = −5 (alıcıya ödeme)

pB = 50

kesikli: sabit nokta (40, 60)

Satıcılar yeni alıcı sayısına tepki verir: nB = 80 + 0,75·25 − 50 = **48,8**. Alıcı sübvansiyonu satıcıyı çekti.

Adım 4 / 14

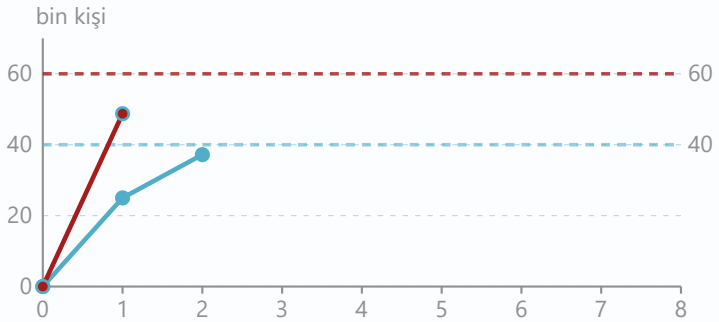
```
1 def katilim(pA, pB, tur=8):
2   nA, nB = 0, 0
3   for t in range(tur):
4     nA = 20 + 0.25 * nB - pA    # alıcılar
▶     nB = 80 + 0.75 * nA - pB    # satıcılar
6   return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `t = 0` `nA = 25.0` `nB = 48.75`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



nA = 37,2

nB = 48,8

pA = −5 (alıcıya ödeme)
pB = 50

kesikli: sabit nokta (40, 60)

Tur 2: alıcılar mevcut satıcı sayısına tepki verir: nA = 20 + 0,25·48,8 + 5 = **37,2**.

Adım 5 / 14

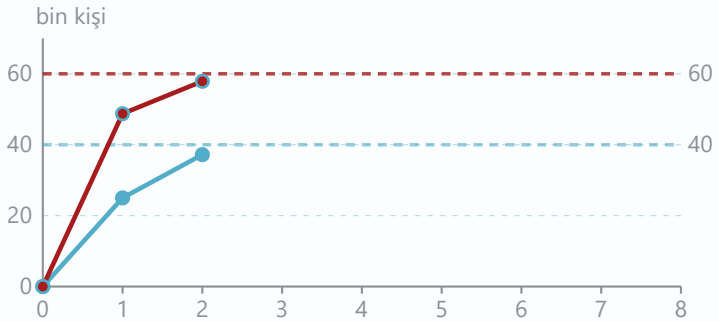
```
1  def katilim(pA, pB, tur=8):
2    nA, nB = 0, 0
3    for t in range(tur):
▶      nA = 20 + 0.25 * nB - pA    # alıcılar
5      nB = 80 + 0.75 * nA - pB    # satıcılar
6    return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `t = 1` `nA = 37.19` `nB = 48.75`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



nA = 37,2

nB = 57,9

pA = −5 (alıcıya ödeme)

pB = 50

kesikli: sabit nokta (40, 60)

Satıcılar yeni alıcı sayısına tepki verir: nB = 80 + 0,75·37,2 − 50 = **57,9**. Çapraz etki iki tarafı birlikte büyütüyor.

Adım 6 / 14

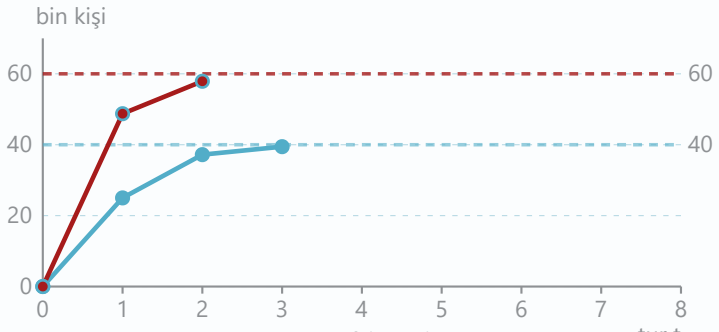
```
1 def katilim(pA, pB, tur=8):
2   nA, nB = 0, 0
3   for t in range(tur):
4     nA = 20 + 0.25 * nB - pA    # alıcılar
▶     nB = 80 + 0.75 * nA - pB    # satıcılar
6   return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `t = 1` `nA = 37.19` `nB = 57.89`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



nA = 39,5

nB = 57,9

pA = −5 (alıcıya ödeme)
pB = 50

kesikli: sabit nokta (40, 60)

Tur 3: alıcılar mevcut satıcı sayısına tepki verir: nA = 20 + 0,25·57,9 + 5 = **39,5**.

Adım 7 / 14

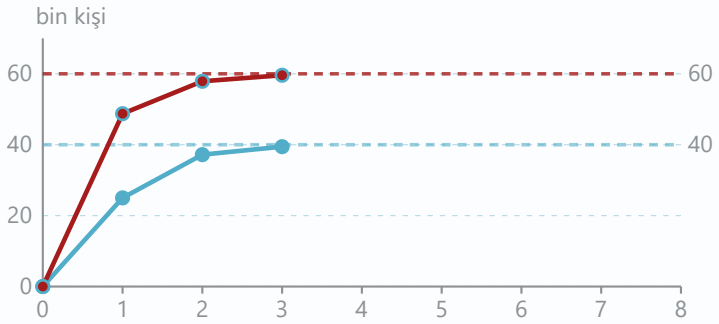
```
1  def katilim(pA, pB, tur=8):
2    nA, nB = 0, 0
3    for t in range(tur):
▶      nA = 20 + 0.25 * nB - pA    # alıcılar
5      nB = 80 + 0.75 * nA - pB    # satıcılar
6    return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `t = 2` `nA = 39.47` `nB = 57.89`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



nA = 39,5

nB = 59,6

pA = −5 (alıcıya ödeme)

pB = 50

kesikli: sabit nokta (40, 60)

Satıcılar yeni alıcı sayısına tepki verir: nB = 80 + 0,75·39,5 − 50 = **59,6**. Çapraz etki iki tarafı birlikte büyütüyor.

Adım 8 / 14

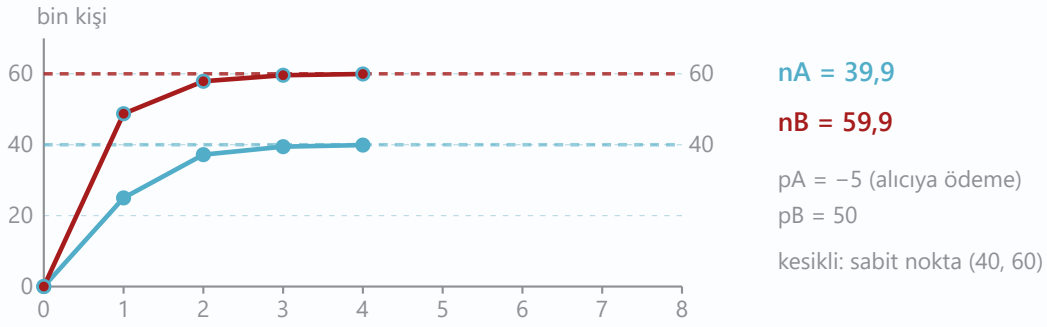
```
1 def katilim(pA, pB, tur=8):
2   nA, nB = 0, 0
3   for t in range(tur):
4     nA = 20 + 0.25 * nB - pA    # alıcılar
▶     nB = 80 + 0.75 * nA - pB    # satıcılar
6   return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `t = 2` `nA = 39.47` `nB = 59.6`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



Tur 4: nA = 39,9, nB = 59,9. Değişim her turda ≈ 0,25·0,75 ≈ 0,19 katına küçülür.

Adım 9 / 14

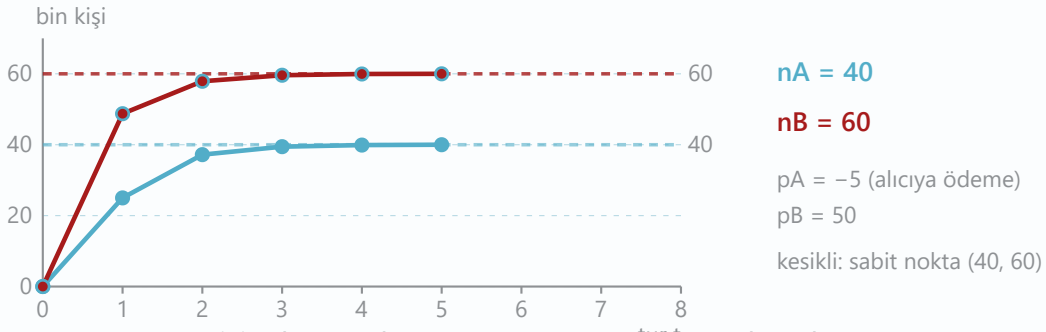
```
1 def katilim(pA, pB, tur=8):
2   nA, nB = 0, 0
3   for t in range(tur):
▶     nA = 20 + 0.25 * nB - pA   # alıcılar
▶     nB = 80 + 0.75 * nA - pB   # satıcılar
6   return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `t = 3` `nA = 39.9` `nB = 59.93`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



Tur 5: nA = 40, nB = 60. Değişim her turda ≈ 0,25·0,75 ≈ 0,19 katına küçülür.

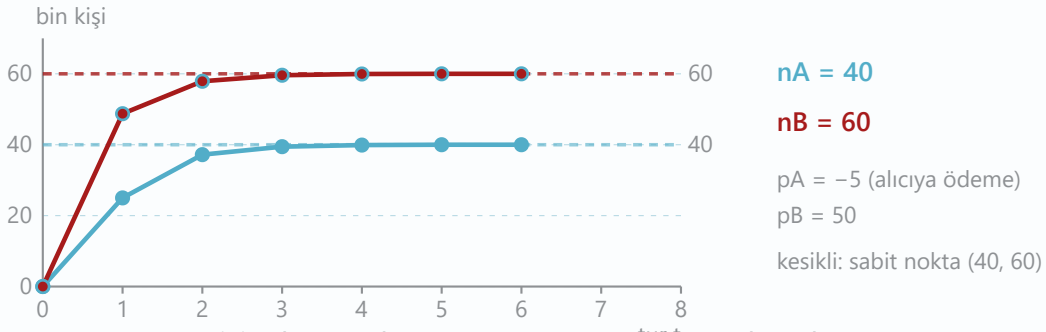

```
1 def katilim(pA, pB, tur=8):
2   nA, nB = 0, 0
3   for t in range(tur):
▶     nA = 20 + 0.25 * nB - pA   # alıcılar
▶     nB = 80 + 0.75 * nA - pB   # satıcılar
6   return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `t = 4` `nA = 39.98` `nB = 59.99`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



Tur 6: nA = 40, nB = 60. Değişim her turda ≈ 0,25·0,75 ≈ 0,19 katına küçülür.

Adım 11 / 14

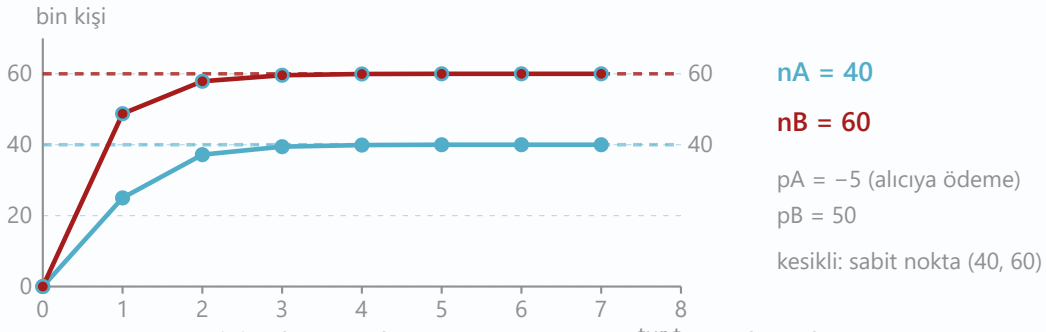
```
1 def katilim(pA, pB, tur=8):
2   nA, nB = 0, 0
3   for t in range(tur):
▶     nA = 20 + 0.25 * nB - pA   # alıcılar
▶     nB = 80 + 0.75 * nA - pB   # satıcılar
6   return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `t = 5` `nA = 40.0` `nB = 60.0`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



Tur 7: nA = 40, nB = 60. Değişim her turda ≈ 0,25·0,75 ≈ 0,19 katına küçülür.

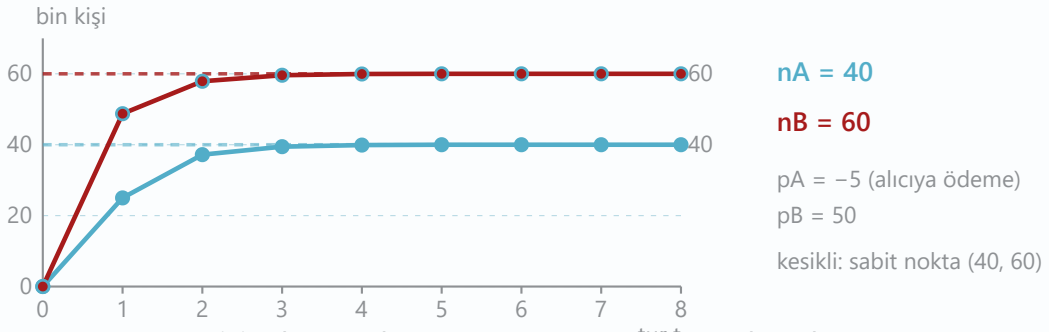

```
1 def katilim(pA, pB, tur=8):
2   nA, nB = 0, 0
3   for t in range(tur):
▶     nA = 20 + 0.25 * nB - pA   # alıcılar
▶     nB = 80 + 0.75 * nA - pB   # satıcılar
6   return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `t = 6` `nA = 40.0` `nB = 60.0`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



Tur 8: nA = 40, nB = 60. Değişim her turda ≈ 0,25·0,75 ≈ 0,19 katına küçülür.

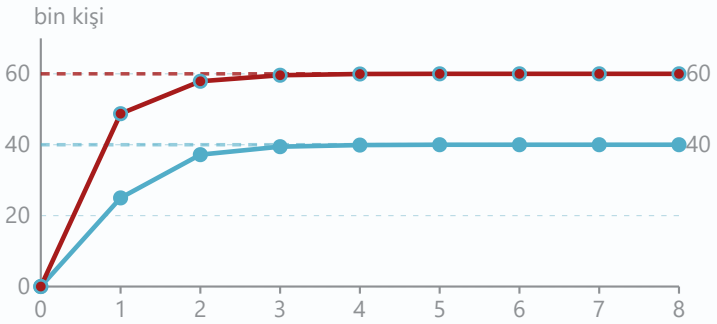

```
1 def katilim(pA, pB, tur=8):
2   nA, nB = 0, 0
3   for t in range(tur):
▶     nA = 20 + 0.25 * nB - pA   # alıcılar
▶     nB = 80 + 0.75 * nA - pB   # satıcılar
6   return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `t = 7` `nA = 40.0` `nB = 60.0`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

Kod İzleme: `katilim(pA, pB)`



nA = 40

nB = 60

pA = −5 (alıcıya ödeme)

pB = 50

kesikli: sabit nokta (40, 60)

Sabit nokta (40, 60). Kâr = −5·40 + 50·60 = **2.800**. Bu, iki tarafın beklentilerinin birbirini doğruladığı katılım dengesidir.

Adım 14 / 14

```
1 def katilim(pA, pB, tur=8):
2   nA, nB = 0, 0
3   for t in range(tur):
4     nA = 20 + 0.25 * nB - pA    # alıcılar
5     nB = 80 + 0.75 * nA - pB    # satıcılar
▶   return nA, nB
```

DEĞİŞKENLER

`pA = -5` `pB = 50` `nA = 40.0` `nB = 60.0`

Yakınsama koşulu: çapraz etkilerin çarpımı $e_A \cdot e_B$ = 0,1875 < 1. Çarpım 1'i aşsaydı katılım patlar ya da çöker: ağ etkisi fazla güçlüdür (devrilme).

PLATFORM FİYATLANDIRMASI

Oyundaki Model: İki Taraflı Platform Fiyatlandırması

Aynı model, iki parametre daha

Oyun: n_A = taban$_A$ − $s_A \cdot p_A + e_A \cdot n_B$, n_B = taban$_B$ − $s_B \cdot p_B + e_B \cdot n_A$.

- taban: ücret 0 ve karşı taraf boşken katılan kullanıcı sayısı.
- s: fiyat duyarlılığı (pazar yerinde $s_A = 11$, $s_B = 5$); sayısal modelde $s = 1$.
- Katılım 0–100 kullanıcıyla sınırlı; ücretler −5 … 10 arasında tam sayı.
- Kâr $\Pi = p_A \cdot n_A + p_B \cdot n_B$, katılım yine karşılıklı tepkinin sabit noktası.

Seviye	e_A / e_B	En kârlı (p_A, p_B)	n_A / n_B	Π
1	0,10 / 0,30	(2, 4)	25,3 / 22,6	140,8
2	0,15 / 0,60	(0, 6)	50,3 / 35,2	211,0
3	0,20 / 0,85	**(−2, 9)**	78,3 / 56,6	352,5

Pazar yeri senaryosu (taban 45 / 35). Seviye 3'te (3, 3) ücretleri yalnız 167,0 kâr getirir: A'yı sübvanse edip B'den yüksek ücret almak kârı iki katından fazlasına çıkarır.

B tarafı A'ya ne kadar çok değer verirse (e_B büyüdükçe) A'nın ücreti o kadar düşer, sonunda negatif olur: p_A = kâr payı − A'nın B tarafına kattığı değer kuralının oyundaki karşılığı.

Sponsorlu Arama Açık Artırması (Sponsored Search)



- Her arama sorgusunda anahtar kelimeye teklif veren reklamverenler için bir açık artırma çalışır.
- m yuva, yukarıdan aşağı tıklanma sayısı $\alpha_1 > \alpha_2 > \dots$ (burada 1000 gösterimde 300, 200, 100).
- Reklamveren i'nin tık başına değeri v_i; tık başına b_i teklif verir, yuva j'de **tık başına p_j** öder.
- Yuva j'deki reklamverenin faydası: $\alpha_j \cdot (v_i - p_j)$.

Örnek: A 10, B 8, C 5, D 2 TL/tık değer (ve teklif). Üç yuva, dört reklamveren.

İki Ödeme Kuralı: GSP ve VCG

GSP (genelleştirilmiş ikinci fiyat)

Teklife göre sırala ($b_{(k)}$: k. en yüksek teklif). Yuva j'deki reklamveren tık başına **bir alttakinin teklifini** öder:

$$p_j = b_{(j+1)}$$

Tek yuvada Vickrey ile aynıdır; birden çok yuvada dürüstlük kaybolur.

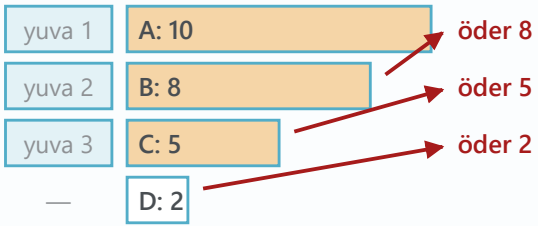
VCG

Aynı sıralama. Reklamveren, varlığıyla **diğerlerine verdiği kaybı** öder:

$$\alpha_j \cdot p_j = \Sigma_{k \geq j} (\alpha_k - \alpha_{k+1}) \cdot b_{(k+1)}$$

Dürüst teklif baskın stratejidir.

GSP: tık başına bir alttakinin teklifi



VCG: A'nın toplam ödemesi = verdiği kayıplar



Yani VCG'de yuva j'deki reklamveren, altındakilerin birer yuva aşağı kaymasıyla kaybettiği tık değerlerinin toplamını öder. $\alpha_j \cdot p_j$ toplam ödeme, p_j tık başı ödeme; $\alpha_{m+1} = 0$.

Adım Adım: Aynı Tekliflerle GSP ve VCG Ödemeleri

Yuva	Tık	Reklamveren	GSP /tık	GSP toplam	VCG toplam	VCG /tık
1	300	A (b = 10)				
2	200	B (b = 8)				
3	100	C (b = 5)				
—	0	D (b = 2)				
		Gelir				

Sıralama teklife göre: A → yuva 1, B → yuva 2, C → yuva 3; D açıkta kalır.

Aynı teklifler (= değerler) iki kurala da verilir; yalnız ödemeler farklıdır.

Adım Adım: Aynı Tekliflerle GSP ve VCG Ödemeleri

Yuva	Tık	Reklamveren	GSP /tık	GSP toplam	VCG toplam	VCG /tık
1	300	A (b = 10)	8	2400		
2	200	B (b = 8)				
3	100	C (b = 5)				
—	0	D (b = 2)	—	0		
		Gelir				

GSP: A, altındaki B'nin teklifini öder: 8 TL/tık × 300 = **2400**.

GSP'de her yuva bir alttaki teklifi, tık başına öder.

Adım Adım: Aynı Tekliflerle GSP ve VCG Ödemeleri

Yuva	Tık	Reklamveren	GSP /tık	GSP toplam	VCG toplam	VCG /tık
1	300	A (b = 10)	8	2400		
2	200	B (b = 8)	5	1000		
3	100	C (b = 5)				
—	0	D (b = 2)	—	0		
		Gelir				

GSP: B, C'nin teklifini öder: 5 × 200 = **1000**.

İkinci yuva.

Adım Adım: Aynı Tekliflerle GSP ve VCG Ödemeleri

Yuva	Tık	Reklamveren	GSP /tık	GSP toplam	VCG toplam	VCG /tık
1	300	A (b = 10)	8	2400		
2	200	B (b = 8)	5	1000		
3	100	C (b = 5)	2	200		
—	0	D (b = 2)	—	0		
		Gelir		**3600**		

GSP: C, D'nin teklifini öder: 2 × 100 = **200**. GSP geliri **3600**.

GSP tamam.

Adım Adım: Aynı Tekliflerle GSP ve VCG Ödemeleri

Yuva	Tık	Reklamveren	GSP /tık	GSP toplam	VCG toplam	VCG /tık
1	300	A (b = 10)	8	2400		
2	200	B (b = 8)	5	1000		
3	100	C (b = 5)	2	200	200	2
—	0	D (b = 2)	—	0	0	—
		Gelir		**3600**		

VCG, C: C olmasaydı D yuva 3'e çıkar: alttakilerin refahı 200; C varken 0 → C öder **200**.

VCG ödemesi = “o olmasaydı” diğerlerinin refahı − o varken diğerlerinin refahı. Üsttekiler etkilenmez; yalnız alttakiler sayılır.

Adım Adım: Aynı Tekliflerle GSP ve VCG Ödemeleri

Yuva	Tık	Reklamveren	GSP /tık	GSP toplam	VCG toplam	VCG /tık
1	300	A (b = 10)	8	2400		
2	200	B (b = 8)	5	1000	700	3,5
3	100	C (b = 5)	2	200	200	2
—	0	D (b = 2)	—	0	0	—
		Gelir		**3600**		

VCG, B: B olmasaydı C yuva 2 (200·5), D yuva 3 (100·2) → 1200; B varken C: 100·5 = 500 → **700**.

B'nin çıkması C'yi bir yuva yükseltir, D'yi yuva 3'e çıkarırdı.

Adım Adım: Aynı Tekliflerle GSP ve VCG Ödemeleri

Yuva	Tık	Reklamveren	GSP /tık	GSP toplam	VCG toplam	VCG /tık
1	300	A (b = 10)	8	2400	1500	5
2	200	B (b = 8)	5	1000	700	3,5
3	100	C (b = 5)	2	200	200	2
—	0	D (b = 2)	—	0	0	—
		Gelir		**3600**	**2400**	

VCG, A: A olmasaydı 300·8 + 200·5 + 100·2 = 3600; A varken 200·8 + 100·5 = 2100 → **1500**.

Aynı sonuç formülden: (300−200)·8 + (200−100)·5 + 100·2 = 1500.

Adım Adım: Aynı Tekliflerle GSP ve VCG Ödemeleri

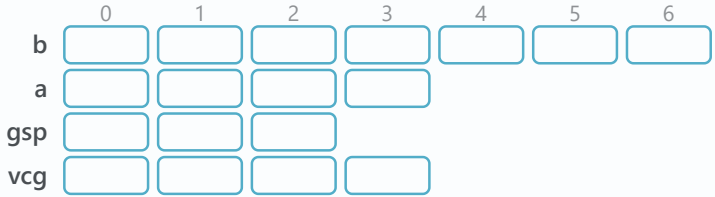
Yuva	Tık	Reklamveren	GSP /tık	GSP toplam	VCG toplam	VCG /tık
1	300	A (b = 10)	8	2400	1500	5
2	200	B (b = 8)	5	1000	700	3,5
3	100	C (b = 5)	2	200	200	2
—	0	D (b = 2)	—	0	0	—
		Gelir		**3600**	**2400**	

Aynı tekliflerde GSP her yuvada VCG'den fazla ya da eşit alır: 8 ≥ 5, 5 ≥ 3,5, 2 ≥ 2 TL/tık.

Gelir: GSP **3600**, VCG **2400**. Ama bu karşılaştırma adil değil: GSP'de teklifler değişir.

SPONSORLU ARAMA

Kod İzleme: `gsp_vcg(bids, ctr)`



b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

m = 3 yuva.

Adım 1 / 14

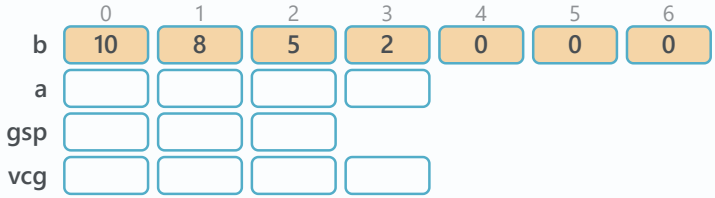
```
1  def gsp_vcg(bids, ctr):
▶     m = len(ctr)
3     b = sorted(bids, reverse=True) + [0]*m
4     a = ctr + [0]      # sanal yuva: 0 tık
5     gsp, vcg = [0]*m, [0]*(m + 1)
6     for j in range(m):
7       gsp[j] = a[j] * b[j+1]   # alttaki
8     for j in range(m-1, -1, -1):  # alttan
9       ek = (a[j] - a[j+1]) * b[j+1]
10      vcg[j] = vcg[j+1] + ek
11    return gsp, vcg[:m]
```

DEĞİŞKENLER

`bids = [5, 10, 2, 8]`

`ctr = [300, 200, 100]` `m = 3`

Kod İzleme: `gsp_vcg(bids, ctr)`



b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

Teklifler büyükten küçüğe sıralanır; sona m sıfır eklenir ki “alttaki teklif” her zaman var olsun.

Adım 2 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
▶    b = sorted(bids, reverse=True) + [0]*m
4    a = ctr + [0]      # sanal yuva: 0 tık
5    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
7      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
9      ek = (a[j] - a[j+1]) * b[j+1]
10     vcg[j] = vcg[j+1] + ek
11   return gsp, vcg[:m]
```

DEĞİŞKENLER

`m = 3` `b = [10, 8, 5, 2, 0, 0, 0]`

Kod İzleme: `gsp_vcg(bids, ctr)`

	0	1	2	3	4	5	6
b	10	8	5	2	0	0	0
a	300	200	100	0			
gsp							
vcg							

b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

a'ya 0 tıklı sanal (m+1). yuva eklenir: $\alpha_{m+1} = 0$.

Adım 3 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
3    b = sorted(bids, reverse=True) + [0]*m
▶    a = ctr + [0]      # sanal yuva: 0 tık
5    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
7      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
9      ek = (a[j] - a[j+1]) * b[j+1]
10     vcg[j] = vcg[j+1] + ek
11   return gsp, vcg[:m]
```

DEĞİŞKENLER

`m = 3` `b = [10, 8, 5, 2, 0, 0, 0]` `a = [300, 200, 100, 0]`

Kod İzleme: `gsp_vcg(bids, ctr)`

	0	1	2	3	4	5	6
b	10	8	5	2	0	0	0
a	300	200	100	0			
gsp	0	0	0				
vcg	0	0	0	0			

b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

Sonuç dizileri sıfırlanır; vcg'nin fazladan son elemanı yinelemenin başlangıç değeridir.

Adım 4 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
3    b = sorted(bids, reverse=True) + [0]*m
4    a = ctr + [0]      # sanal yuva: 0 tık
▶    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
7      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
9      ek = (a[j] - a[j+1]) * b[j+1]
10     vcg[j] = vcg[j+1] + ek
11   return gsp, vcg[:m]
```

DEĞİŞKENLER

`m = 3` `b = [10, 8, 5, 2, 0, 0, 0]` `a = [300, 200, 100, 0]` `gsp = [0, 0, 0]` `vcg = [0, 0, 0, 0]`

Kod İzleme: `gsp_vcg(bids, ctr)`

	0	1	2	3	4	5	6
b	10	8	5	2	0	0	0
a	300	200	100	0			
gsp	2400	0	0				
vcg	0	0	0	0			

j = 0

b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

gsp[0] = a[0]·b[1] = 300·8 = **2400**: yuva 1, altındakinin teklifini tık başına öder.

Adım 5 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
3    b = sorted(bids, reverse=True) + [0]*m
4    a = ctr + [0]      # sanal yuva: 0 tık
5    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
▶      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
9      ek = (a[j] - a[j+1]) * b[j+1]
10     vcg[j] = vcg[j+1] + ek
11   return gsp, vcg[:m]
```

DEĞİŞKENLER

`m = 3` `b = [10, 8, 5, 2, 0, 0, 0]` `a = [300, 200, 100, 0]` `gsp = [2400, 0, 0]` `vcg = [0, 0, 0, 0]` `j = 0`

Kod İzleme: `gsp_vcg(bids, ctr)`

	0	1	2	3	4	5	6
b	10	8	5	2	0	0	0
a	300	200	100	0			
gsp	2400	1000	0				
vcg	0	0	0	0			

j = 1

b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

gsp[1] = a[1]·b[2] = 200·5 = **1000**: yuva 2, altındakinin teklifini tık başına öder.

Adım 6 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
3    b = sorted(bids, reverse=True) + [0]*m
4    a = ctr + [0]      # sanal yuva: 0 tık
5    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
▶      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
9      ek = (a[j] - a[j+1]) * b[j+1]
10     vcg[j] = vcg[j+1] + ek
11   return gsp, vcg[:m]
```

DEĞİŞKENLER

`m = 3` `b = [10, 8, 5, 2, 0, 0, 0]`

`a = [300, 200, 100, 0]`

`gsp = [2400, 1000, 0]`

`vcg = [0, 0, 0, 0]` `j = 1`

Kod İzleme: `gsp_vcg(bids, ctr)`

	0	1	2	3	4	5	6
b	10	8	5	2	0	0	0
a	300	200	100	0			
gsp	2400	1000	200				
vcg	0	0	0	0			

j = 2

b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

gsp[2] = a[2]·b[3] = 100·2 = **200**: yuva 3, altındakinin teklifini tık başına öder.

Adım 7 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
3    b = sorted(bids, reverse=True) + [0]*m
4    a = ctr + [0]      # sanal yuva: 0 tık
5    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
▶      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
9      ek = (a[j] - a[j+1]) * b[j+1]
10     vcg[j] = vcg[j+1] + ek
11   return gsp, vcg[:m]
```

DEĞİŞKENLER

`m = 3` `b = [10, 8, 5, 2, 0, 0, 0]` `a = [300, 200, 100, 0]` `gsp = [2400, 1000, 200]` `vcg = [0, 0, 0, 0]` `j = 2`

Kod İzleme: `gsp_vcg(bids, ctr)`

	0	1	2	3	4	5	6
b	10	8	5	2	0	0	0
a	300	200	100	0			
gsp	2400	1000	200				
vcg	0	0	0	0			

j = 2

b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

ek = (100 − 0)·2 = **200**: alttaki reklamveren yuva 3 yerine 4'te kaldığı için kaybettiği tık değeri.

Adım 8 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
3    b = sorted(bids, reverse=True) + [0]*m
4    a = ctr + [0]      # sanal yuva: 0 tık
5    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
7      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
▶      ek = (a[j] - a[j+1]) * b[j+1]
10     vcg[j] = vcg[j+1] + ek
11   return gsp, vcg[:m]
```

DEĞİŞKENLER

`m = 3` `b = [10, 8, 5, 2, 0, 0, 0]` `a = [300, 200, 100, 0]` `gsp = [2400, 1000, 200]` `vcg = [0, 0, 0, 0]` `j = 2` `ek = 200`

Kod İzleme: `gsp_vcg(bids, ctr)`

	0	1	2	3	4	5	6
b	10	8	5	2	0	0	0
a	300	200	100	0			
gsp	2400	1000	200				
vcg	0	0	200	0			

j = 2

b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

vcg[2] = 0 + 200 = **200**: alttakilere verilen kayıp da üste taşınır.

Adım 9 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
3    b = sorted(bids, reverse=True) + [0]*m
4    a = ctr + [0]      # sanal yuva: 0 tık
5    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
7      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
9      ek = (a[j] - a[j+1]) * b[j+1]
▶      vcg[j] = vcg[j+1] + ek
11   return gsp, vcg[:m]
```

DEĞİŞKENLER

`m = 3` `b = [10, 8, 5, 2, 0, 0, 0]`

`a = [300, 200, 100, 0]`

`gsp = [2400, 1000, 200]`

`vcg = [0, 0, 200, 0]` `j = 2` `ek = 200`

Kod İzleme: `gsp_vcg(bids, ctr)`

	0	1	2	3	4	5	6
b	10	8	5	2	0	0	0
a	300	200	100	0			
gsp	2400	1000	200				
vcg	0	0	200	0			

j = 1

b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

ek = (200 − 100)·5 = **500**: alttaki reklamveren yuva 2 yerine 3'te kaldığı için kaybettiği tık değeri.

Adım 10 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
3    b = sorted(bids, reverse=True) + [0]*m
4    a = ctr + [0]      # sanal yuva: 0 tık
5    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
7      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
▶      ek = (a[j] - a[j+1]) * b[j+1]
10     vcg[j] = vcg[j+1] + ek
11   return gsp, vcg[:m]
```

DEĞİŞKENLER

`m = 3` `b = [10, 8, 5, 2, 0, 0, 0]` `a = [300, 200, 100, 0]` `gsp = [2400, 1000, 200]` `vcg = [0, 0, 200, 0]` `j = 1` `ek = 500`

Kod İzleme: `gsp_vcg(bids, ctr)`

	0	1	2	3	4	5	6
b	10	8	5	2	0	0	0
a	300	200	100	0			
gsp	2400	1000	200				
vcg	0	700	200	0			

j = 1

b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

vcg[1] = 200 + 500 = **700**: alttakilere verilen kayıp da üste taşınır.

Adım 11 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
3    b = sorted(bids, reverse=True) + [0]*m
4    a = ctr + [0]      # sanal yuva: 0 tık
5    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
7      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
9      ek = (a[j] - a[j+1]) * b[j+1]
▶      vcg[j] = vcg[j+1] + ek
11   return gsp, vcg[:m]
```

DEĞİŞKENLER

`m = 3` `b = [10, 8, 5, 2, 0, 0, 0]`

`a = [300, 200, 100, 0]`

`gsp = [2400, 1000, 200]`

`vcg = [0, 700, 200, 0]` `j = 1`

`ek = 500`

Kod İzleme: `gsp_vcg(bids, ctr)`

	0	1	2	3	4	5	6
b	10	8	5	2	0	0	0
a	300	200	100	0			
gsp	2400	1000	200				
vcg	0	700	200	0			

j = 0

b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

ek = (300 − 200)·8 = **800**: alttaki reklamveren yuva 1 yerine 2'de kaldığı için kaybettiği tık değeri.

Adım 12 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
3    b = sorted(bids, reverse=True) + [0]*m
4    a = ctr + [0]      # sanal yuva: 0 tık
5    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
7      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
▶      ek = (a[j] - a[j+1]) * b[j+1]
10     vcg[j] = vcg[j+1] + ek
11   return gsp, vcg[:m]
```

DEĞİŞKENLER

`m = 3` `b = [10, 8, 5, 2, 0, 0, 0]`

`a = [300, 200, 100, 0]`

`gsp = [2400, 1000, 200]`

`vcg = [0, 700, 200, 0]` `j = 0`

`ek = 800`

Kod İzleme: `gsp_vcg(bids, ctr)`

	0	1	2	3	4	5	6
b	10	8	5	2	0	0	0
a	300	200	100	0			
gsp	2400	1000	200				
vcg	1500	700	200	0			

j = 0

b: sıralı teklifler (TL/tık)

a: yuva tıklanmaları

gsp[j]: toplam GSP ödemesi

vcg[j]: toplam VCG ödemesi

vcg[0] = 700 + 800 = **1500**: alttakilere verilen kayıp da üste taşınır.

Adım 13 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
3    b = sorted(bids, reverse=True) + [0]*m
4    a = ctr + [0]      # sanal yuva: 0 tık
5    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
7      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
9      ek = (a[j] - a[j+1]) * b[j+1]
▶      vcg[j] = vcg[j+1] + ek
11   return gsp, vcg[:m]
```

DEĞİŞKENLER

`m = 3` `b = [10, 8, 5, 2, 0, 0, 0]`

`a = [300, 200, 100, 0]`

`gsp = [2400, 1000, 200]`

`vcg = [1500, 700, 200, 0]` `j = 0`

`ek = 800`

Kod İzleme: `gsp_vcg(bids, ctr)`

	0	1	2	3	4	5	6	
b	10	8	5	2	0	0	0	b: sıralı teklifler (TL/tık)
a	300	200	100	0				a: yuva tıklanmaları
gsp	2400	1000	200					gsp[j]: toplam GSP ödemesi
vcg	1500	700	200	0				vcg[j]: toplam VCG ödemesi

Sonuç: GSP [2400, 1000, 200] (toplam 3600), VCG [1500, 700, 200] (toplam 2400).

Adım 14 / 14

```
1  def gsp_vcg(bids, ctr):
2    m = len(ctr)
3    b = sorted(bids, reverse=True) + [0]*m
4    a = ctr + [0]     # sanal yuva: 0 tık
5    gsp, vcg = [0]*m, [0]*(m + 1)
6    for j in range(m):
7      gsp[j] = a[j] * b[j+1]   # alttaki
8    for j in range(m-1, -1, -1):  # alttan
9      ek = (a[j] - a[j+1]) * b[j+1]
10     vcg[j] = vcg[j+1] + ek
▶    return gsp, vcg[:m]
```

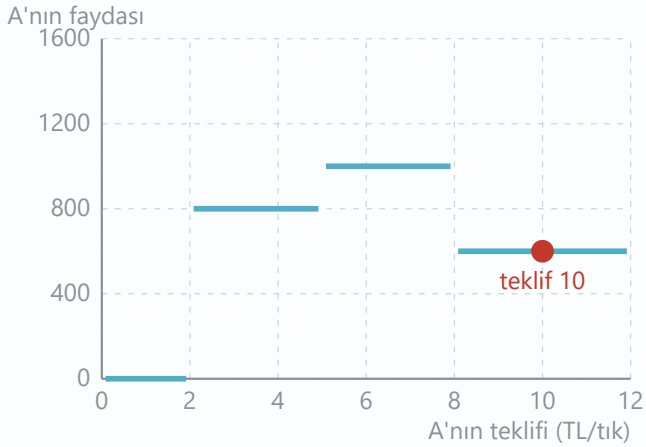
DEĞİŞKENLER

`gsp = [2400, 1000, 200]`

`vcg = [1500, 700, 200]`

Adım Adım: GSP'de Dürüst Teklif Denge Değildir

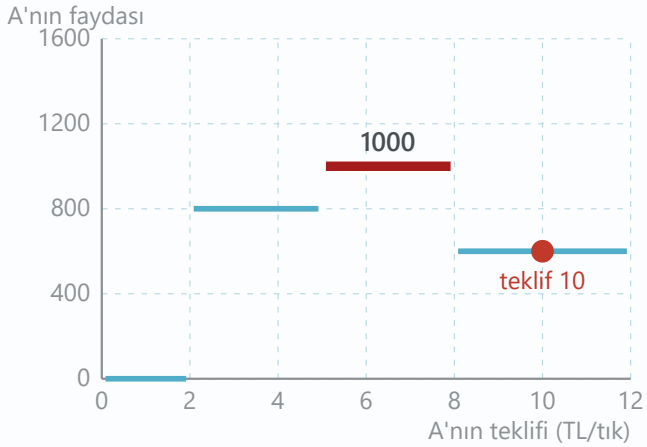
Rek.	v	teklif	yuva	öder/tık	fayda
A	10	10	1	8	**600**
B	8	8	2	5	**600**
C	5	5	3	2	**300**
D	2	2	—	—	**0**
Platform geliri					**3600**



Herkes değerini teklif ediyor. A yuva 1'de, B'nin teklifini (8) öder: fayda 300·(10 − 8) = **600**.

Adım Adım: GSP'de Dürüst Teklif Denge Değildir

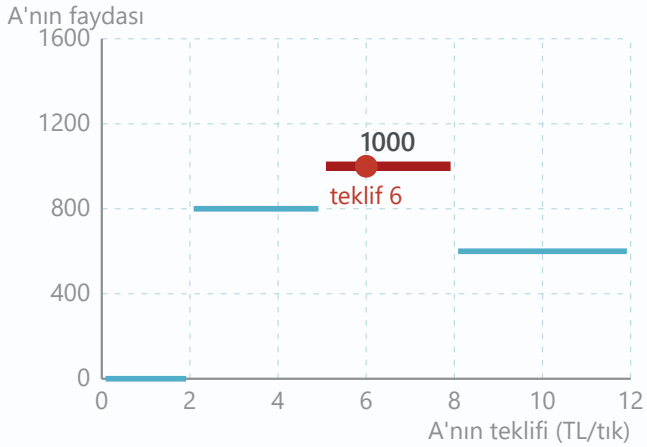
Rek.	v	teklif	yuva	öder/tık	fayda
A	10	10	1	8	**600**
B	8	8	2	5	**600**
C	5	5	3	2	**300**
D	2	2	—	—	**0**
Platform geliri					**3600**



A'nın faydası teklifine göre basamaklı: 2–5 arası yuva 3 (800), **5–8 arası yuva 2 (1000)**, 8 üstü yuva 1 (600).

Adım Adım: GSP'de Dürüst Teklif Denge Değildir

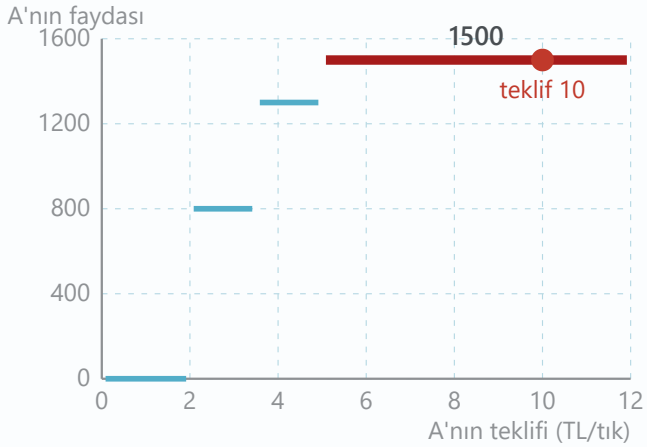
Rek.	v	teklif	yuva	öder/tık	fayda
B	8	8	1	6	**600**
A	10	6	2	5	**1000**
C	5	5	3	2	**300**
D	2	2	—	—	**0**
Platform geliri					**3000**



A 6 teklif ederse yuva 2'ye iner, C'nin 5'ini öder: 200·(10 − 5) = **1000 > 600**. Dürüst teklif en iyi tepki değil → **denge değil**.

Adım Adım: GSP'de Dürüst Teklif Denge Değildir

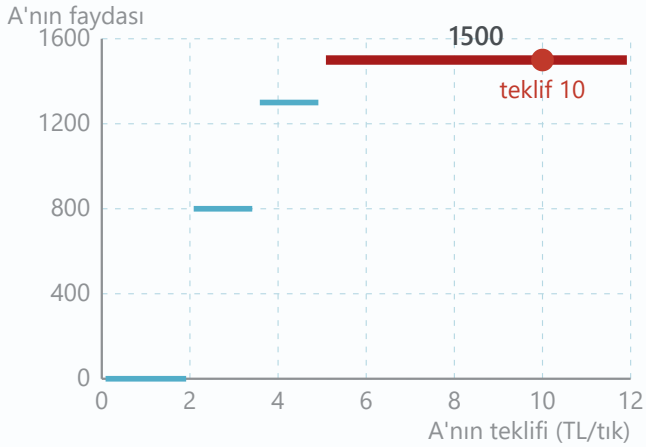
Rek.	v	teklif	yuva	öder/tık	fayda
A	10	10	1	5	**1500**
B	8	5	2	3,5	**900**
C	5	3,5	3	2	**300**
D	2	2	—	—	**0**
Platform geliri					**2400**



Bir denge: B 5, C 3,5, D 2 teklif eder. A için en iyisi yine yuva 1 (1500); betik tüm sapmaları denedi, kimse kazanamıyor.

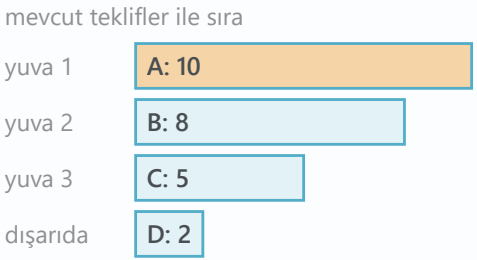
Adım Adım: GSP'de Dürüst Teklif Denge Değildir

Rek.	v	teklif	yuva	öder/tık	fayda
A	10	10	1	5	**1500**
B	8	5	2	3,5	**900**
C	5	3,5	3	2	**300**
D	2	2	—	—	**0**
Platform geliri					**2400**



Bu dengede tık başı ödemeler 5 / 3,5 / 2 = **VCG ödemeleri**, gelir 2400. GSP'nin kıskançsız (envy-free) dengeleri içinde en düşük gelirli olan budur. Kıskançsız denge: hiçbir reklamveren başka bir yuvayı o yuvanın şu anki tık başı fiyatıyla almayı kendi yerine tercih etmez.

Kod İzleme: `karli_sapma(i, b, v, ctr, adaylar)`



v_A = 10 · tıklar 300/200/100

Dürüst teklifler b = v = [10, 8, 5, 2]. A (i = 0) sapmalı mı? Aday teklifler 9, 6, 4, 3, 1.

Adım 1 / 12

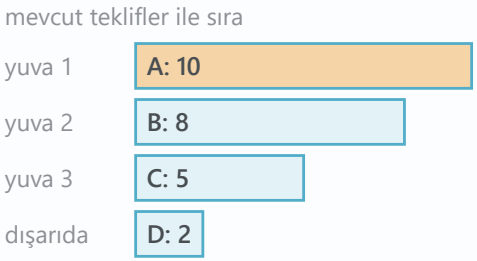
```
▶ def karli_sapma(i, b, v, ctr, adaylar):
2   eski = gsp_fayda(i, b, v, ctr)
3   for x in adaylar:
4     b2 = b[:]; b2[i] = x   # yalnız i değişir
5     u = gsp_fayda(i, b2, v, ctr)
6     if u > eski:
7       return x, u   # kârlı sapma
8   return None     # sapma yok
```

DEĞİŞKENLER

`i = 0` `b = [10, 8, 5, 2]`

gsp_fayda(i, b, v, ctr): teklifleri sırala; i, j. yuvadaysa ctr[j]·(v[i] − bir alttaki teklif), yuva dışındaysa 0. Teklifler farklı varsayılır.

Kod İzleme: `karli_sapma(i, b, v, ctr, adaylar)`



v_A = 10 · tıklar 300/200/100

eski fayda = 600

Mevcut durumda A yuva 1'de, bir alttaki teklifi (8) öder: eski = 300·(10 − 8) = **600**.

Adım 2 / 12

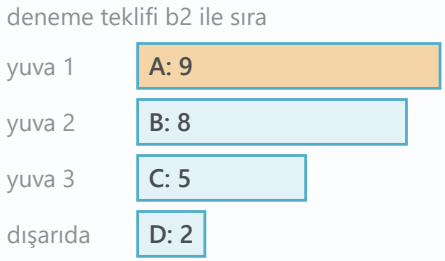
```
1 def karli_sapma(i, b, v, ctr, adaylar):
▶   eski = gsp_fayda(i, b, v, ctr)
3   for x in adaylar:
4     b2 = b[:]; b2[i] = x   # yalnız i değişir
5     u = gsp_fayda(i, b2, v, ctr)
6     if u > eski:
7       return x, u   # kârlı sapma
8   return None     # sapma yok
```

DEĞİŞKENLER

`i = 0` `b = [10, 8, 5, 2]` `eski = 600`

gsp_fayda(i, b, v, ctr): teklifleri sırala; i, j. yuvadaysa ctr[j]·(v[i] − bir alttaki teklif), yuva dışındaysa 0. Teklifler farklı varsayılır.

Kod İzleme: `karli_sapma(i, b, v, ctr, adaylar)`



v_A = 10 · tıklar 300/200/100

eski fayda = 600

x = 9 → yuva 1, u = 600

u > eski değil → devam

x = 9: yuva 1, u = 600 ≤ 600.

Adım 3 / 12

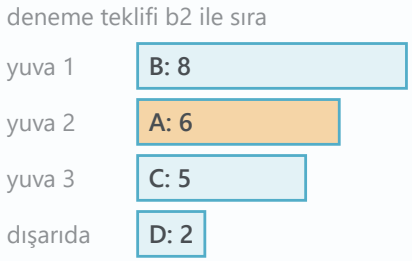
```
1 def karli_sapma(i, b, v, ctr, adaylar):
2   eski = gsp_fayda(i, b, v, ctr)
▶   for x in adaylar:
▶     b2 = b[:]; b2[i] = x   # yalnız i değişir
▶     u = gsp_fayda(i, b2, v, ctr)
▶     if u > eski:
7       return x, u   # kârlı sapma
8   return None     # sapma yok
```

DEĞİŞKENLER

`i = 0` `b = [10, 8, 5, 2]` `eski = 600` `x = 9` `u = 600`

gsp_fayda(i, b, v, ctr): teklifleri sırala; i, j. yuvadaysa ctr[j]·(v[i] − bir alttaki teklif), yuva dışındaysa 0. Teklifler farklı varsayılır.

Kod İzleme: `karli_sapma(i, b, v, ctr, adaylar)`



v_A = 10 · tıklar 300/200/100

eski fayda = 600

x = 6 → yuva 2, u = 1.000

u > eski → kârlı sapma!

döndür: (6, 1.000)

x = 6: A yuva 2'ye iner, daha ucuz öder: u = **1.000** > 600 → **kârlı sapma bulundu**, (x, u) döner.

Adım 4 / 12

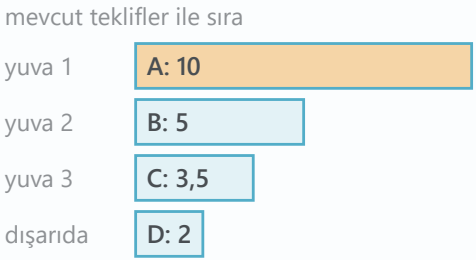
```
1 def karli_sapma(i, b, v, ctr, adaylar):
2   eski = gsp_fayda(i, b, v, ctr)
▶   for x in adaylar:
▶     b2 = b[:]; b2[i] = x   # yalnız i değişir
▶     u = gsp_fayda(i, b2, v, ctr)
▶     if u > eski:
▶       return x, u   # kârlı sapma
8   return None      # sapma yok
```

DEĞİŞKENLER

`i = 0` `b = [10, 8, 5, 2]` `eski = 600` `x = 6` `u = 1000`

gsp_fayda(i, b, v, ctr): teklifleri sırala; i, j. yuvadaysa ctr[j]·(v[i] − bir alttaki teklif), yuva dışındaysa 0. Teklifler farklı varsayılır.

Kod İzleme: `karli_sapma(i, b, v, ctr, adaylar)`



v_A = 10 · tıklar 300/200/100

Şimdi önerilen denge: B 5, C 3,5, D 2 teklif ediyor. A için aynı aday listesi.

Adım 5 / 12

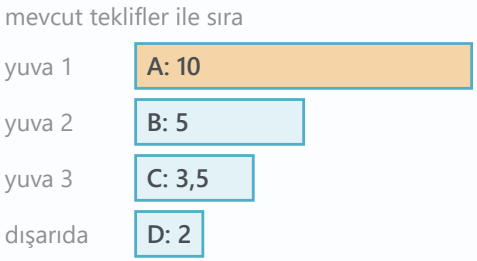
```
▶ def karli_sapma(i, b, v, ctr, adaylar):
2   eski = gsp_fayda(i, b, v, ctr)
3   for x in adaylar:
4     b2 = b[:]; b2[i] = x   # yalnız i değişir
5     u = gsp_fayda(i, b2, v, ctr)
6     if u > eski:
7       return x, u   # kârlı sapma
8   return None     # sapma yok
```

DEĞİŞKENLER

`i = 0` `b = [10, 5, 3.5, 2]`

gsp_fayda(i, b, v, ctr): teklifleri sırala; i, j. yuvadaysa ctr[j]·(v[i] − bir alttaki teklif), yuva dışındaysa 0. Teklifler farklı varsayılır.

Kod İzleme: `karli_sapma(i, b, v, ctr, adaylar)`



v_A = 10 · tıklar 300/200/100

eski fayda = 1.500

Mevcut durumda A yuva 1'de, bir alttaki teklifi (5) öder: eski = 300·(10 − 5) = **1.500**.

Adım 6 / 12

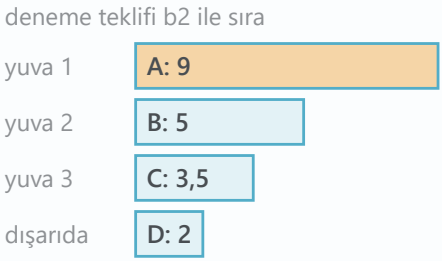
```
1 def karli_sapma(i, b, v, ctr, adaylar):
▶   eski = gsp_fayda(i, b, v, ctr)
3   for x in adaylar:
4     b2 = b[:]; b2[i] = x   # yalnız i değişir
5     u = gsp_fayda(i, b2, v, ctr)
6     if u > eski:
7       return x, u   # kârlı sapma
8   return None     # sapma yok
```

DEĞİŞKENLER

`i = 0` `b = [10, 5, 3.5, 2]` `eski = 1500`

gsp_fayda(i, b, v, ctr): teklifleri sırala; i, j. yuvadaysa ctr[j]·(v[i] − bir alttaki teklif), yuva dışındaysa 0. Teklifler farklı varsayılır.

Kod İzleme: `karli_sapma(i, b, v, ctr, adaylar)`



v_A = 10 · tıklar 300/200/100

eski fayda = 1.500

x = 9 → yuva 1, u = 1.500

u > eski değil → devam

x = 9: yuva 1, u = 1.500 ≤ 1.500.

Adım 7 / 12

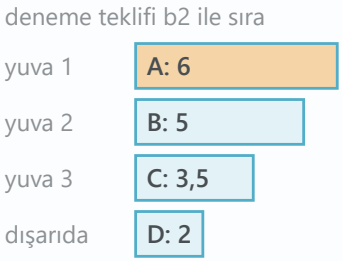
```
1 def karli_sapma(i, b, v, ctr, adaylar):
2   eski = gsp_fayda(i, b, v, ctr)
▶   for x in adaylar:
▶     b2 = b[:]; b2[i] = x   # yalnız i değişir
▶     u = gsp_fayda(i, b2, v, ctr)
▶     if u > eski:
7       return x, u   # kârlı sapma
8   return None     # sapma yok
```

DEĞİŞKENLER

`i = 0` `b = [10, 5, 3.5, 2]` `eski = 1500` `x = 9` `u = 1500`

gsp_fayda(i, b, v, ctr): teklifleri sırala; i, j. yuvadaysa ctr[j]·(v[i] − bir alttaki teklif), yuva dışındaysa 0. Teklifler farklı varsayılır.

Kod İzleme: `karli_sapma(i, b, v, ctr, adaylar)`



v_A = 10 · tıklar 300/200/100

eski fayda = 1.500

x = 6 → yuva 1, u = 1.500

u > eski değil → devam

x = 6: yuva 1, u = 1.500 ≤ 1.500.

Adım 8 / 12

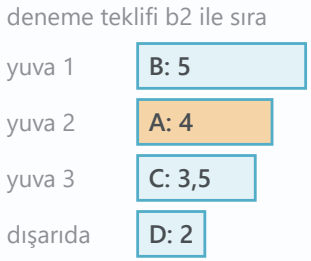
```
1 def karli_sapma(i, b, v, ctr, adaylar):
2   eski = gsp_fayda(i, b, v, ctr)
▶   for x in adaylar:
▶     b2 = b[:]; b2[i] = x   # yalnız i değişir
▶     u = gsp_fayda(i, b2, v, ctr)
▶     if u > eski:
7       return x, u   # kârlı sapma
8   return None     # sapma yok
```

DEĞİŞKENLER

`i = 0` `b = [10, 5, 3.5, 2]` `eski = 1500` `x = 6` `u = 1500`

gsp_fayda(i, b, v, ctr): teklifleri sırala; i, j. yuvadaysa ctr[j]·(v[i] − bir alttaki teklif), yuva dışındaysa 0. Teklifler farklı varsayılır.

Kod İzleme: `karli_sapma(i, b, v, ctr, adaylar)`



v_A = 10 · tıklar 300/200/100

eski fayda = 1.500

x = 4 → yuva 2, u = 1.300

u > eski değil → devam

x = 4: yuva 2, u = 1.300 ≤ 1.500.

Adım 9 / 12

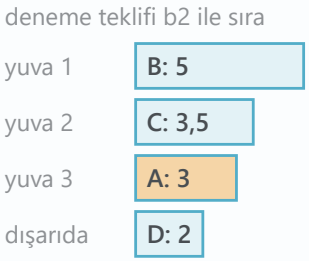
```
1 def karli_sapma(i, b, v, ctr, adaylar):
2   eski = gsp_fayda(i, b, v, ctr)
▶   for x in adaylar:
▶     b2 = b[:]; b2[i] = x   # yalnız i değişir
▶     u = gsp_fayda(i, b2, v, ctr)
▶     if u > eski:
7       return x, u   # kârlı sapma
8   return None     # sapma yok
```

DEĞİŞKENLER

`i = 0` `b = [10, 5, 3.5, 2]` `eski = 1500` `x = 4` `u = 1300`

gsp_fayda(i, b, v, ctr): teklifleri sırala; i, j. yuvadaysa ctr[j]·(v[i] − bir alttaki teklif), yuva dışındaysa 0. Teklifler farklı varsayılır.

Kod İzleme: `karli_sapma(i, b, v, ctr, adaylar)`



v_A = 10 · tıklar 300/200/100

eski fayda = 1.500

x = 3 → yuva 3, u = 800

u > eski değil → devam

x = 3: yuva 3, u = 800 ≤ 1.500.

Adım 10 / 12

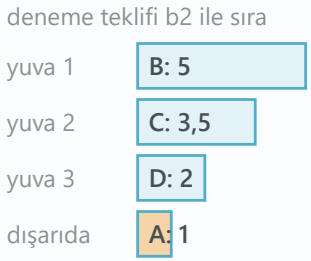
```
1 def karli_sapma(i, b, v, ctr, adaylar):
2   eski = gsp_fayda(i, b, v, ctr)
▶   for x in adaylar:
▶     b2 = b[:]; b2[i] = x   # yalnız i değişir
▶     u = gsp_fayda(i, b2, v, ctr)
▶     if u > eski:
7       return x, u   # kârlı sapma
8   return None     # sapma yok
```

DEĞİŞKENLER

`i = 0` `b = [10, 5, 3.5, 2]` `eski = 1500` `x = 3` `u = 800`

gsp_fayda(i, b, v, ctr): teklifleri sırala; i, j. yuvadaysa ctr[j]·(v[i] − bir alttaki teklif), yuva dışındaysa 0. Teklifler farklı varsayılır.

Kod İzleme: `karli_sapma(i, b, v, ctr, adaylar)`



v_A = 10 · tıklar 300/200/100

eski fayda = 1.500

x = 1 → yuva —, u = 0

u > eski değil → devam

x = 1: reklam dışarıda kalır, u = 0 ≤ 1.500.

Adım 11 / 12

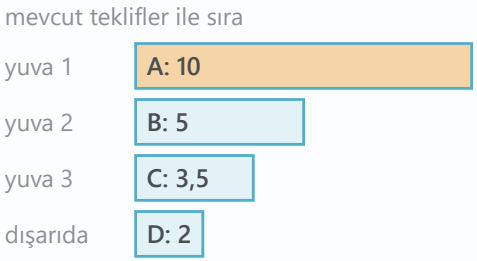
```
1 def karli_sapma(i, b, v, ctr, adaylar):
2   eski = gsp_fayda(i, b, v, ctr)
▶   for x in adaylar:
▶     b2 = b[:]; b2[i] = x   # yalnız i değişir
▶     u = gsp_fayda(i, b2, v, ctr)
▶     if u > eski:
7       return x, u   # kârlı sapma
8   return None     # sapma yok
```

DEĞİŞKENLER

`i = 0` `b = [10, 5, 3.5, 2]` `eski = 1500` `x = 1` `u = 0`

gsp_fayda(i, b, v, ctr): teklifleri sırala; i, j. yuvadaysa ctr[j]·(v[i] − bir alttaki teklif), yuva dışındaysa 0. Teklifler farklı varsayılır.

Kod İzleme: `karli_sapma(i, b, v, ctr, adaylar)`



v_A = 10 · tıklar 300/200/100

eski fayda = 1.500

döndür: None

Hiçbir aday kârlı değil → **None**. Betik aynı testi dört reklamverenin hepsi için 0,25 adımlı teklif ızgarasında yaptı: kimse kazanamıyor, bu profil bir Nash dengesi.

Adım 12 / 12

```
1 def karli_sapma(i, b, v, ctr, adaylar):
2   eski = gsp_fayda(i, b, v, ctr)
3   for x in adaylar:
4     b2 = b[:]; b2[i] = x   # yalnız i değişir
5     u = gsp_fayda(i, b2, v, ctr)
6     if u > eski:
7       return x, u   # kârlı sapma
▶   return None     # sapma yok
```

DEĞİŞKENLER

`i = 0` `b = [10, 5, 3.5, 2]` `eski = 1500`

gsp_fayda(i, b, v, ctr): teklifleri sırala; i, j. yuvadaysa ctr[j]·(v[i] − bir alttaki teklif), yuva dışındaysa 0. Teklifler farklı varsayılır.

GSP ve VCG: Karşılaştırma

	GSP	VCG
Dürüst teklif baskın mı?	Hayır (A 10 yerine 6 verir)	Evet
Aynı tekliflerle gelir	3600 (her yuvada ≥ VCG)	2400
Dengede gelir	Yerel kıskançsız dengelerde ≥ 2400; en düşüğü VCG ile aynı	2400 (dürüst teklifle)
Açıklaması	Kolay: "bir alttakinin teklifi"	Dışsallık toplamı, daha zor
Uygulama	Arama reklamlarında uzun süre yaygın (kalite puanıyla)	Facebook reklam açık artırmasında kullanıldı (Varian ve Harris, 2014)

Kalite puanı

Pratikte sıralamada teklifle birlikte reklamın tıklanma kalitesi (kalite puanı) de kullanılır.

Ders

"İkinci fiyat" adı dürüstlüğü garanti etmez: tek nesnede geçerli olan, çok yuvada bozulabilir.

SPONSORLU ARAMA

Verili Teklif Profili ≠ Denge Teklifleri

GSP sonuçları verili tekliflerle hesaplanır. 3600, herkesin dürüst teklif ettiği profildeki gelirdir. Ama dürüst teklif GSP'de baskın strateji değildir. Tık oranları 300, 200, 100; değerler A 10, B 8, C 5, D 2.

Profil	Teklifler (A, B, C, D)	A'nın yuvası ve faydası	Platform geliri
Dürüst	10, 8, 5, 2	yuva 1: 300·(10 − 8) = 600	3600
A daha düşük teklif verir	6, 8, 5, 2	yuva 2: 200·(10 − 5) = **1000**	3000
Yerel kıskançsız denge	10, 5, 3,5, 2	yuva 1: 300·(10 − 5) = 1500	**2400** = VCG

- A daha alt yuvada daha az tık alır ama tık başına daha az öder: 1000 > 600.
- **Yerel kıskançsız denge** (Edelman, Ostrovsky, Schwarz, 2007): hiçbir reklamveren bir üstündeki yuvayı o yuvanın şu anki fiyatıyla almak istemez. B yuva 2'de 200·(8 − 3,5) = 900 alır; yuva 1'i 5'ten alsa 300·(8 − 5) = 900 alırdı: kazancı yok.

Bu dengelerde GSP geliri en az 2400'dür; en düşüğü VCG ile aynıdır. Dürüst tekliflerle bulunan 3600, dengedeki gelir değildir.

SPONSORLU ARAMA

Bütçeler ve Çevrim İçi Tahsis

- Reklamverenlerin günlük **bütçesi** vardır; aramalar tek tek gelir ve platform her birini geleceği bilmeden hemen bir reklamverene verir (*çevrim içi* karar).
- Örnek: A ve B'nin bütçesi 100 TL. "koşu ayakkabısı" için ikisi de 1 TL, "spor çanta" için yalnız A 1 TL teklif verir. Önce 100 ayakkabı, sonra 100 çanta araması gelir.

Kural	Ayakkabı aramaları	Çanta aramaları	Gelir
Açgözlü (en yüksek teklif, eşitlikte A)	100'ü A'ya; A'nın bütçesi biter	A'nın parası yok: 0	100
Dengeleme (Balance): kalan bütçesi büyük olana	A 50, B 50	A'nın kalan 50'si	150
Geleceği bilen optimum	100'ü B'ye	100'ü A'ya	**200**

Sık hata

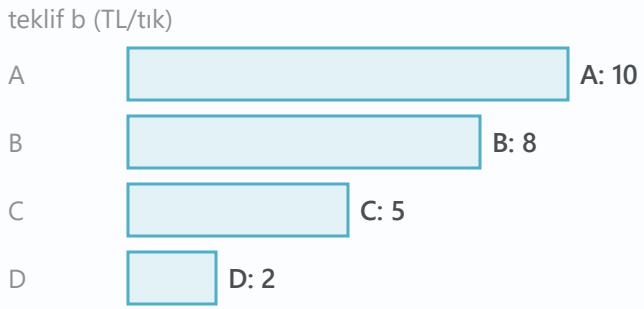
"Her aramada en yüksek teklifi seçmek en iyisidir." Bütçe varken açgözlü kural, esnek reklamvereni erkenden tüketir; en kötü durumda optimumun yalnız 1/2'sini garanti eder.

Garanti

Kalan bütçeyi gözeten kurallar, arama sırası ne olursa olsun optimumun en az $1 - 1/e \approx$ %63'ünü elde eder (Mehta, Saberi, Vazirani ve Vazirani, 2007).

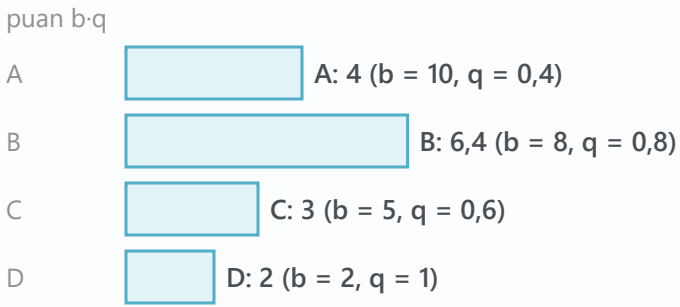
SPONSORLU ARAMA

Adım Adım: Kalite Puanıyla GSP



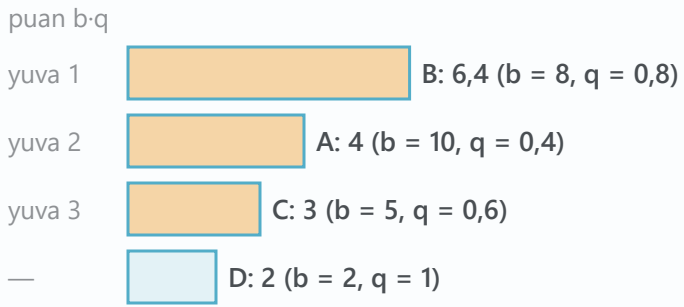
Aynı dört reklamveren. Tık sayısı yuva ve reklam kalitesinin çarpımı: yuva j'de i'nin tık sayısı $\alpha_j \cdot q_i$ (α = 300, 200, 100).

Adım Adım: Kalite Puanıyla GSP



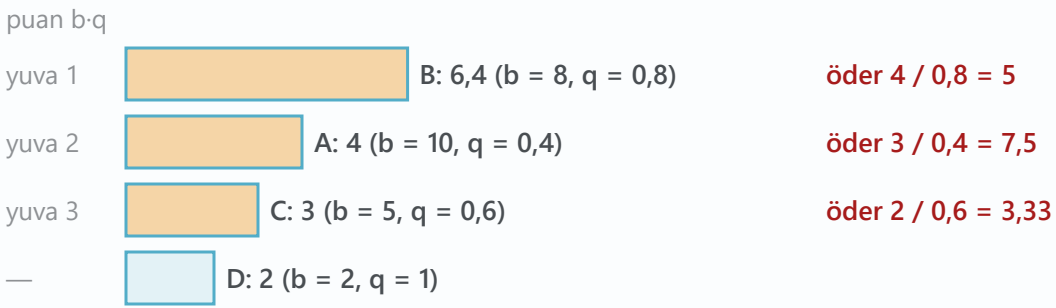
Kalite q: reklamın tıklanma eğilimi (ör. A 0,4, B 0,8). Platform **b·q** puanına göre sıralar: bir gösterimin beklenen tık başı geliri.

Adım Adım: Kalite Puanıyla GSP



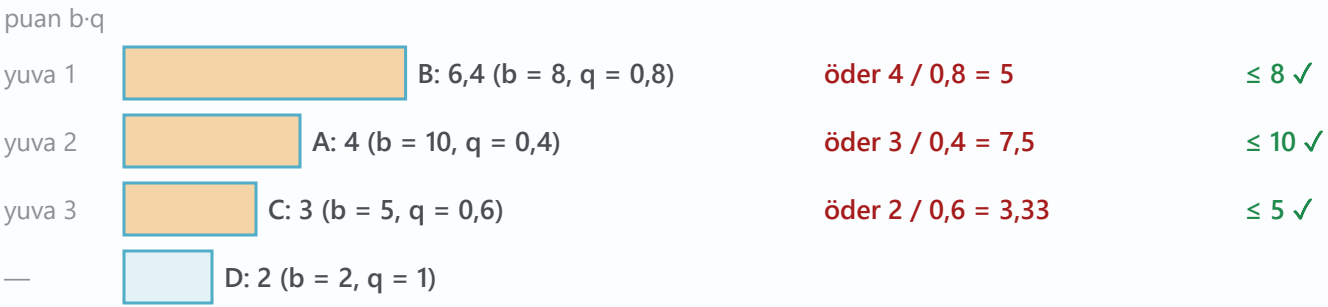
Puana göre sıra B (6,4) > A (4) > C (3) > D (2). En yüksek teklif veren A artık birinci değil.

Adım Adım: Kalite Puanıyla GSP



Ödeme: yerini korumak için gereken en düşük teklif. $b_i q_i \geq$ alttakinin puanı $\iff b_i \geq \text{puan}_{j+1}/q_i$.

Adım Adım: Kalite Puanıyla GSP



Her ödeme teklifi aşmaz (5 ≤ 8, 7,5 ≤ 10, 3,33 ≤ 5). Kaliteli reklam aynı yuvaya daha ucuza çıkar: kalite ödüllendirilir.

Adım Adım: Kalite Puanıyla GSP

Sıralama	yuva 1–3	toplam tık (1000 gösterimde)	gelir (TL)
Yalnız teklife göre	A, B, C	340	1.880
Kalite puanıyla (b·q)	B, A, C	380	2.000

Bu örnekte kalite puanı hem tıkları (340 → 380) hem geliri (1.880 → 2.000) artırdı: kullanıcı ilgili reklamı görür, platform daha çok tık satar.

SPONSORLU ARAMA

Oyuna Dönüş: Sponsorlu Arama Açık Artırması

Kuramın cevabı: hayır

"Search Ads" oranlarıyla (%12, %7, %3,5) 1000 gösterimde 120, 70, 35 tık. Tık değeriniz 8, rakip teklifleri 6, 4, 2, kaliteler 1; GSP (Seviye 2):

Teklif	Yuva	Ödeme/tık	Fayda
8 (dürüst)	1	6	120 × 2 = 240
5	2	4	**70 × 4 = 280**
3	3	2	35 × 6 = 210

Alt yuvaya inmek daha kârlı; oyunun "Find best response" düğmesi bu sapmayı arar.

Gerçekte

Birinci fiyatlı açık artırmada (Seviye 1'in kuralı) teklifler sürekli yükselip çöktü (Edelman ve Ostrovsky, 2007). Google'ın kullandığı GSP'de de dürüst teklif baskın değildir (Edelman, Ostrovsky ve Schwarz, 2007).

"İkinci fiyat" adı tek yuvada dürüstlüğü getirir; çok yuvada getirmez. Dürüstlüğü garanti eden kural VCG'dir.

İTİBAR VE REKABET

İtibar Sistemleri ve Bilgi Asimetrisi

- Alıcı, satıcının kalitesini önceden bilmez: bilgi asimetrisi. Kalite bilinmeyince alıcı ortalama fiyat önerir, iyi satıcı çekilir, piyasada kötü ürünler kalır (Akerlof'un “limonlar piyasası”).
- İtibar sistemi iki iş görür: **seçim** (iyi satıcıyı bul) ve **disiplin** (kötü davranışı gelecekte cezalandır, tekrarlanan oyun).

Sahte değerlendirme
Satın alınmış olumlu yorum veya rakibe olumsuz yorum.

Misilleme korkusu
Karşılıklı puanlamada kötü deneyim bildirilmez.

Not enflasyonu
Herkes 4,8+ ise puan ayırt edici olmaz.

Seçici katılım
Orta düzeyde memnun kalanlar sessiz kalır; yeni satıcı hiç yorum alamaz (soğuk başlangıç).

Tasarım önlemleri: doğrulanmış satın alma şartı, eski yorumlara daha az ağırlık, az yorumlu puanın belirsizliğini hesaba katmak (Bayesçi ortalama), karşılıklı puanlamada iki tarafın puanını aynı anda açıklamak.

Adım Adım: Az Yorumlu 5,0 mı, Çok Yorumlu 4,7 mi?

Ürün	yorum	ortalama	sıra	Bayesçi puan	sıra
X	2	5	1		
Y	200	4,7	2		
Z	20	4,2	3		

Ham ortalamaya göre sıra: X (5,0) > Y (4,7) > Z (4,2). Ama X'in yalnız 2 yorumu var!

Az yorumlu ürünün ortalaması çok gürültülüdür; iki sahte yorum onu zirveye taşır.

Adım Adım: Az Yorumlu 5,0 mı, Çok Yorumlu 4,7 mi?

Ürün	yorum	ortalama	sıra	Bayesçi puan	sıra
X	2	5	1	3,75	3
Y	200	4,7	2	4,64	1
Z	20	4,2	3	3,97	2

Bayesçi ortalama: $(C \cdot m + \Sigma \text{ puan})/(C + n)$, $m = 3{,}5$ (platform ortalaması), $C = 10$, n: ürünün yorum sayısı. X: $(35 + 10)/12 = 3{,}75$; Y: $(35 + 940)/210 \approx 4{,}64$.

Her ürüne 10 tane “ortalama” sanal yorum eklenmiş gibi: az yorumlu ürün ortalamaya çekilir. Yeni sıra: Y > Z > X.

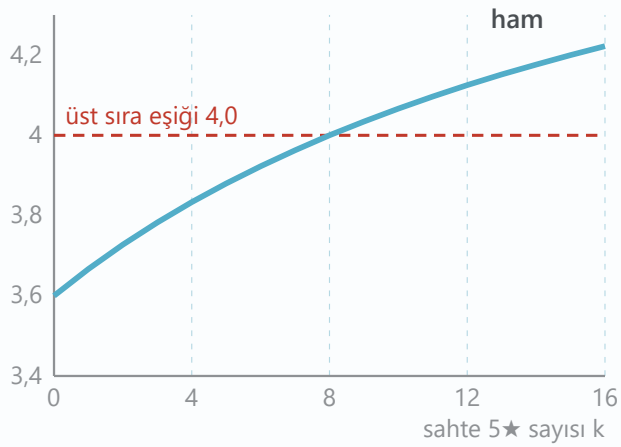
Adım Adım: Az Yorumlu 5,0 mı, Çok Yorumlu 4,7 mi?

Ürün	yorum	ortalama	sıra	Bayesçi puan	sıra
X + 3 sahte	5	5	1	4	3
Y	200	4,7	2	4,64	1
Z	20	4,2	3	3,97	2

X'e 3 sahte 5★ eklense ham ortalama yine 5,0 (zirvede kalır); Bayesçi puan (35 + 25)/15 = 4,0 → hâlâ Y'nin altında.

Bayesçi ortalama sahte yorumların etkisini azaltır: aynı etkiyi yaratmak için çok daha fazla sahte yorum gerekir.

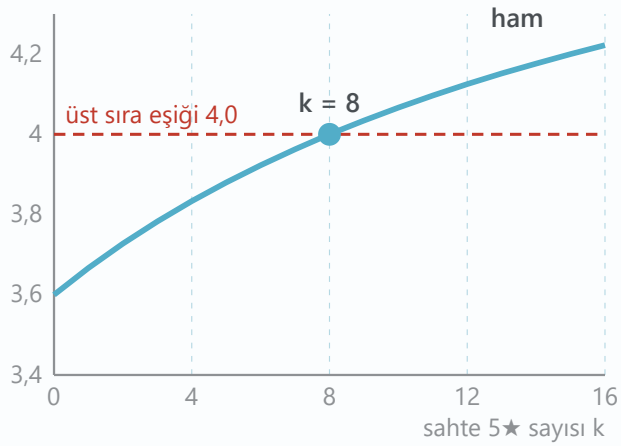
Adım Adım: Sahte Değerlendirme Ne Zaman Kârlı?



Satıcı: 20 yorum, ortalama 3,6. Puanı ≥ 4,0 olanlar üst sırada → aylık ek kâr G = 600 TL. Sahte 5★ yorumun tanesi 15 TL.

Sahte yorumu bir yatırım kararı olarak modelle: getiri, maliyet ve beklenen ceza.

Adım Adım: Sahte Değerlendirme Ne Zaman Kârlı?

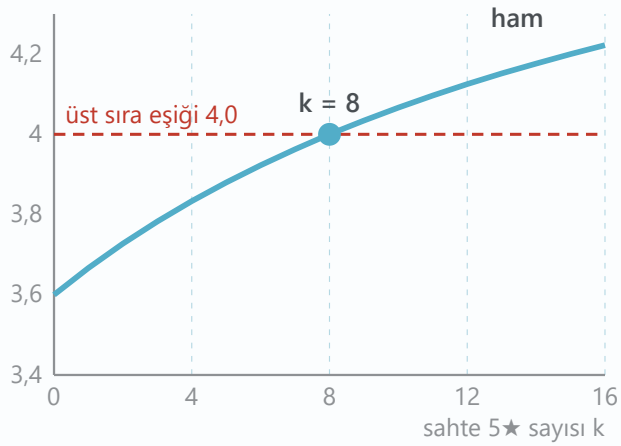


Satıcı: 20 yorum, ortalama 3,6. Puanı ≥ 4,0 olanlar üst sırada → aylık ek kâr G = 600 TL. Sahte 5★ yorumun tanesi 15 TL.

Ham ortalama: $(72 + 5k)/(20 + k) \geq 4 \iff k \geq 8$. Maliyet $8 \times 15 = 120$ TL.

8 sahte 5★ yorum ham ortalamayı tam 4,0'a çıkarır.

Adım Adım: Sahte Değerlendirme Ne Zaman Kârlı?



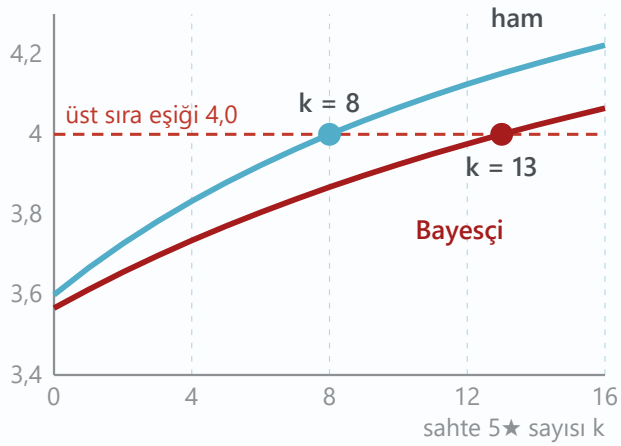
Satıcı: 20 yorum, ortalama 3,6. Puanı ≥ 4,0 olanlar üst sırada → aylık ek kâr G = 600 TL. Sahte 5★ yorumun tanesi 15 TL.

Ham ortalama: $(72 + 5k)/(20 + k) \geq 4 \iff k \geq 8$. Maliyet $8 \times 15 = 120$ TL.

Tespit olasılığı ρ, yakalanınca kayıp L = 3000 TL. Kârlı $\iff 600 - 120 > \rho \cdot 3000 \iff$ **$\rho < 0{,}16$**.

Beklenen kazanç $600 - 120 - \rho \cdot 3000$. Platformun tespit olasılığı %16'nın altındaysa hile kârlı.

Adım Adım: Sahte Değerlendirme Ne Zaman Kârlı?



Satıcı: 20 yorum, ortalama 3,6. Puanı ≥ 4,0 olanlar üst sırada → aylık ek kâr G = 600 TL. Sahte 5★ yorumun tanesi 15 TL.

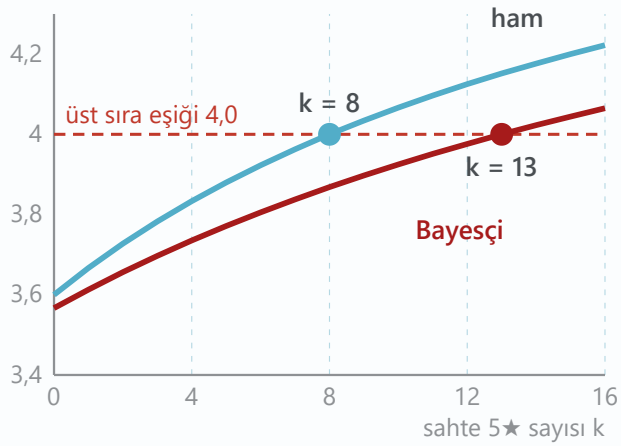
Ham ortalama: $(72 + 5k)/(20 + k) \geq 4 \iff k \geq 8$. Maliyet $8 \times 15 = 120$ TL.

Tespit olasılığı ρ, yakalanınca kayıp L = 3000 TL. Kârlı $\iff 600 - 120 > \rho \cdot 3000 \iff$ **$\rho < 0{,}16$**.

Bayesçi puanla (m = 3,5, C = 10): $(107 + 5k)/(30 + k) \geq 4 \iff k \geq 13$; maliyet 195 TL → **$\rho < 0{,}135$**.

Aynı eşik Bayesçi puanda 13 sahte yorum ister: hile pahalanır, kârlılık sınırı $\rho < 0{,}135$'e iner.

Adım Adım: Sahte Değerlendirme Ne Zaman Kârlı?



Satıcı: 20 yorum, ortalama 3,6. Puanı ≥ 4,0 olanlar üst sırada → aylık ek kâr G = 600 TL. Sahte 5★ yorumun tanesi 15 TL.

Ham ortalama: $(72 + 5k)/(20 + k) \geq 4 \iff k \geq 8$. Maliyet $8 \times 15 = 120$ TL.

Tespit olasılığı ρ, yakalanınca kayıp L = 3000 TL. Kârlı $\iff 600 - 120 > \rho \cdot 3000 \iff$ **$\rho < 0{,}16$**.

Bayesçi puanla (m = 3,5, C = 10): $(107 + 5k)/(30 + k) \geq 4 \iff k \geq 13$; maliyet 195 TL → **$\rho < 0{,}135$**.

Tasarım kolları: ρ'yu artır (doğrulanmış alım, anormallik tespiti), L'yi artır (hesap kapatma), G'yi azalt (sağlam puanlama).

Mekanizma tasarımı bakışı: platform hileyi yasaklamakla yetinmez, **hilenin beklenen getirisini negatife** çeker.

Platform Rekabeti ve Düzenleme

Rekabet

- **Kazanan her şeyi alır** (winner-takes-all): devrilme nedeniyle tek platform pazarın çoğunu alabilir.
- **Geçiş maliyeti**: arkadaşlar, geçmiş puanlar, veriler → kullanıcıyı elde tutar.
- **Çoklu kullanım** (multi-homing): aynı anda birden çok platform (sürücü hem Uber hem rakip) tekelleşmeyi zayıflatır.
- Örnekler: Hepsiburada – Trendyol, Uber – Lyft, Instagram – TikTok.

Etik ve düzenleme

- Veri mahremiyeti ve güvenlik sorunları.
- Tekelleşme riski ve kendi ürününü kayırma (self-preferencing).
- Sıralama ve öneri algoritmalarının şeffaflığı.
- Örnek: Avrupa Birliği Dijital Piyasalar Yasası (Digital Markets Act, DMA) büyük "geçit bekçisi" platformlara yükümlülükler getirir.

İTİBAR VE REKABET

Öneri Sistemleri, Bağlılık ve Mahremiyet

- Öneri ve sıralama algoritması **görünürlüğü tahsis eder** (kim öne çıkacak?): bir tahsis kuralıdır, katılımcılar stratejilerini ona göre seçer.

Platformun hedefi

Kısa vadeli etkileşim ya da gelir maksimize edilirse kullanıcının uzun vadeli yararıyla çatışabilir (tıklama tuzağı, sonsuz akış).

Üreticiler de oyuncu

Üreticiler ve satıcılar görünürlük için yarışır, algoritmanın ödüllendirdiği biçime (başlık, etiket) yönelir: içerik tekdüzeleşebilir.

Bağlılık programları

Sadakat puanı, ücretli üyelik (ör. hızlı kargo), oyunlaştırma (rozet, seri): geçiş maliyetini bilerek artırır, çoklu kullanımı azaltır.

Veri ve mahremiyet

"Ücretsiz" hizmetin bedeli veridir: daha çok veri daha iyi öneri ama sızıntı ve kötüye kullanım riski. Kullanıcı ne kadar paylaşacağını da stratejik seçer.

Mekanizma tasarımı sorusu yine aynı: platformun amacı ile kullanıcıların ve üreticilerin çıkarı uyumlu mu?

Tartışma: Büyüme, Davranış ve Adalet

Neden bazıları hızla büyür, bazıları çöker?

Kritik kitle ve devrilme: beklenti eşiğin üstündeyse geri besleme büyütür, altındaysa çökertir. Kolay kullanım, güven ve erken teşvikler eşiği aşmayı kolaylaştırır.

Platformlar davranışı nasıl şekillendirir?

Bonus ve indirimler belirli davranışları artırır (yoğun bölgede sürücü bonusu); öneri algoritmaları tercihleri yönlendirir; spam ve sahte yorum denetimi manipülasyonu sınırlar.

Tasarım herkese adil olabilir mi?

Tam adalet zordur: farklı tercihler, ağ etkisinin büyük grupları kayırması, stratejik manipülasyon. Teşvik, kural ve düzenleme ile iyileştirilebilir.

DEĞERLENDİRME

Özet

Kavram	Açıklama	Örnek
Ağ etkisi, devrilme	$F(x) = 1 - p/x$; iki kararlı denge ve aradaki eşik	$p = 0{,}16$: 0 ve 0,8 kararlı, eşik 0,2
İki taraflı fiyat	$p_A = n_A - e_B n_B$: diğer tarafa çok değer katan ucuzlar	$p_A = -5$, $p_B = 50$, $\Pi = 2800$
GSP	Yuva j, bir alttakinin teklifini öder; dürüst teklif baskın değil	aynı tekliflerle gelir 3600; A 10 yerine 6 verir
VCG	Diğerlerine verilen kaybı öder; dürüst teklif baskın	1500 / 700 / 200, gelir 2400
GSP dengesi	Kıskançsız dengelerin en düşüğü VCG sonucunu verir	teklifler 10, 5, 3,5, 2 → 5 / 3,5 / 2 TL/tık
İtibar	Bayesçi ortalama, sahte yorumu pahalılaştırır	gereken sahte yorum 8 → 13

DEĞERLENDİRME

Örnek Problemler

1. Benimseme modelinde $p = 0{,}21$ olsun. Dengeler ve kritik kitle nedir?
2. Pazar yeri modelinde $e_A = e_B = 0{,}5$ (diğer her şey aynı). Optimum fiyatlar nedir? Sübvansiyon var mı?
3. İki yuva (100 ve 60 tık), teklifler 6, 4, 1. GSP ve VCG toplam ödemelerini bulun.
4. 3. sorudaki GSP'de değeri 6 olan reklamveren, diğerleri dürüstken dürüst teklif vermeli mi?
5. 10 yorum, ortalama 4,0. Ham ortalamayı 4,5'e çıkarmak için kaç sahte 5★ gerekir?
6. İki yuva; A 6 ($q = 0{,}5$), B 4 ($q = 1$), C 1 ($q = 1$) teklif ediyor. Kalite puanlı GSP'de sıra ve tık başı ödemeler nedir?

▶ **Cevapları göster**

DEĞERLENDİRME

Etkileşimli Oyunlar

Sponsorlu Arama Açık Artırması

Kalite puanı, çok yuvalı tıklanma oranları, birinci fiyat ve GSP ödemeleri; en iyi tepki arayışı.

İki Taraflı Platform Fiyatlandırması

İki tarafın ücretine göre katılım, çapraz ağ etkisi ve kâr. Hangi taraf sübvanse edilmeli?

DEĞERLENDİRME

Kaynaklar

- J.-C. Rochet, J. Tirole, "Platform Competition in Two-Sided Markets", *Journal of the European Economic Association*, 2003.
- M. Armstrong, "Competition in Two-Sided Markets", *RAND Journal of Economics*, 2006.
- B. Edelman, M. Ostrovsky, M. Schwarz, "Internet Advertising and the Generalized Second-Price Auction", *American Economic Review*, 2007.
- H. Varian, "Position Auctions", *International Journal of Industrial Organization*, 2007.
- D. Easley, J. Kleinberg, ***Networks, Crowds, and Markets***, Cambridge University Press, 2010 (Bölüm 15 ve 17).
- N. Nisan, T. Roughgarden, É. Tardos, V. Vazirani (ed.), *Algorithmic Game Theory*, Cambridge University Press, 2007 (Bölüm 27–28).
- T. Roughgarden, **Stanford CS364A: Algorithmic Game Theory** ders notları, 2013 (2.–3. dersler: sponsorlu arama açık artırmaları).
- G. Akerlof, "The Market for 'Lemons'", *Quarterly Journal of Economics*, 1970.
- M. Luca, G. Zervas, "Fake It Till You Make It: Reputation, Competition, and Yelp Review Fraud", *Management Science*, 2016.

DEĞERLENDİRME

Kaynaklar: Reklam Açık Artırmaları

- H. Varian, C. Harris, "The VCG Auction in Theory and Practice", *American Economic Review: Papers & Proceedings* 104(5), 2014.
- B. Edelman, M. Ostrovsky, "Strategic Bidder Behavior in Sponsored Search Auctions", *Decision Support Systems* 43(1), 2007. · M. Ostrovsky, M. Schwarz, "Reserve Prices in Internet Advertising Auctions: A Field Experiment", *ACM EC*, 2011.
- A. Mehta, A. Saberi, U. Vazirani, V. Vazirani, "AdWords and Generalized Online Matching", *Journal of the ACM* 54(5), 2007.

Son

Bölüm 6: Çevrim İçi Platformlar

Sonraki bölüm: Bölüm 7: Sosyal Seçim ve Oylama

Çalışma soruları · Terim sözlüğü