Web Advertising

Rafael Ferreira da Silva
rafsilva@isi.edu
http://rafaelsilva.com

https://xkcd.com/870/

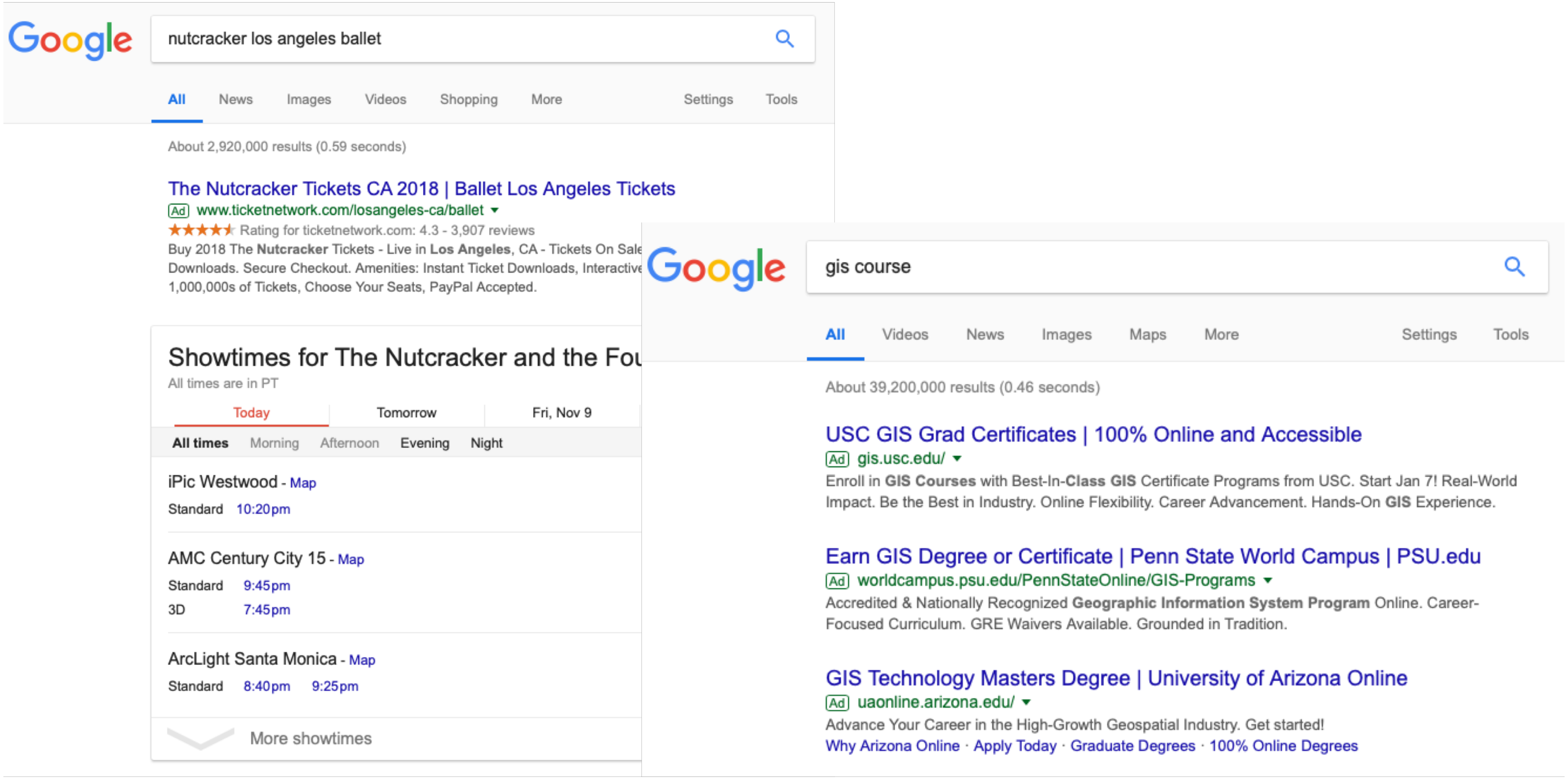
Chapter 8 Overview

- Ability of all sorts of Web applications to support themselves through advertising
- **Most lucrative venue for on-line advertising: SEARCH**
- **Adwords model (Google): matching search queries to advertisements**
  - Algorithms for optimizing this assignment
  - **Greedy** algorithms
  - **Online** algorithms
- Selecting items to advertise at an on-line store
  - Use collaborative filtering

Types of Web Ads

- **Advertisers post ads directly**
  - Craig's List, auto trading sites
- **Advertisers pay for display ads to be placed on Web sites**
  - Fixed price per ***impression*** (one display of the ad with download of page by a user)
- **Online stores show ads**
  - Amazon, Macy's, etc.
  - Not paid for by manufacturers of product advertised
  - **Selected by store to maximize probability customer will buy product**
  - **Collaborative Filtering**
- **Search ads are placed among results of a search query**
  - Advertisers bid for right to have their ad shown in response to certain queries
  - **Pay only if ad is clicked on**

Search Ads



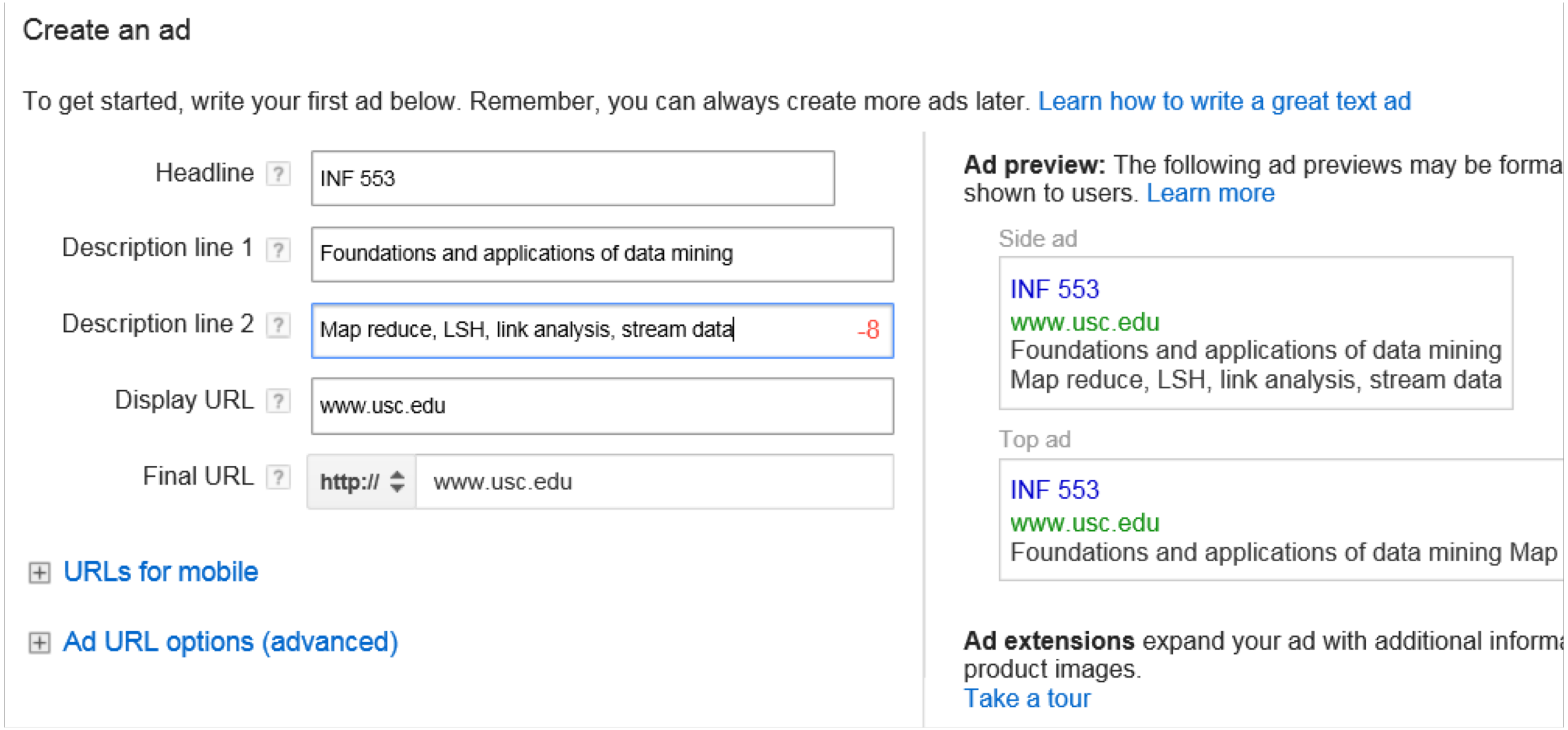
USC Viterbi School of Engineering Information Sciences Institute

Search Ads

- Impression of an Ad
  - Ad is displayed
  - A user clicked on the ad link to download the page

- Search engine charges advertisers for impression of their ads

Google AdWords



USC Viterbi School of Engineering Information Sciences Institute

Google AdWords

⊟ Select keywords

Your ad can show on Google when people search for the keywords you choose here. These keywords also determine which managed placements are good matches for your ads.

Tips

- Start with 10-20 keywords.
- Be specific: avoid one-word keywords. Choose phrases that customers would use to search for your products and services.
- By default, keywords are broad matched to searches to help you capture a wider range of relevant traffic. Use match types to control this.
- Learn more about choosing effective keywords.

Enter one keyword per line.

```
usc informatics
usc data mining
```

⊟ Category: University Of Southern California

« Add all from this category

« Add southern california university
« Add california southern university
« Add the university of southern california
« Add southern university of california
« Add universities in southern california
« Add university of southern california campus
« Add university southern california
« Add university in southern california
« Add where is university of southern california
« Add southern california universities
« Add where is the university of southern

USC Viterbi School of Engineering
Information Sciences Institute

Matching Keywords with Searches

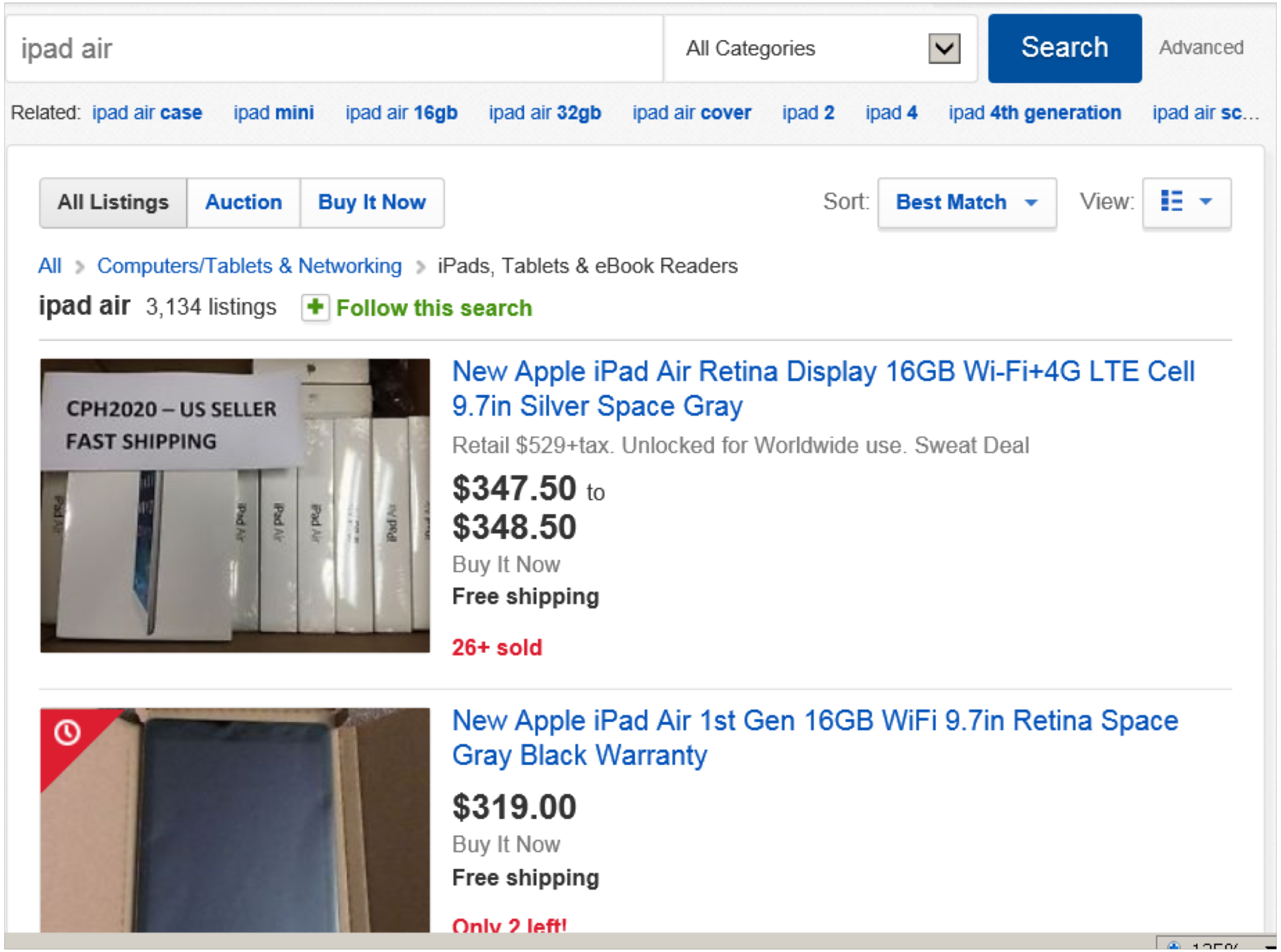
- Match types: exact, phrase, broad, negative

Match Type Examples

Match type	Example keyword	Example search	What searched can be matched?
Broad	women's hats	*buy ladies hats*	Misspellings, synonyms, etc
Phrase	“women's hats”	*buy women's hats*	Include phrase or its close variation
Exact	[women's hats]	women's hats	Exact term or close variation
Negative	-women	*men hats*	Without term

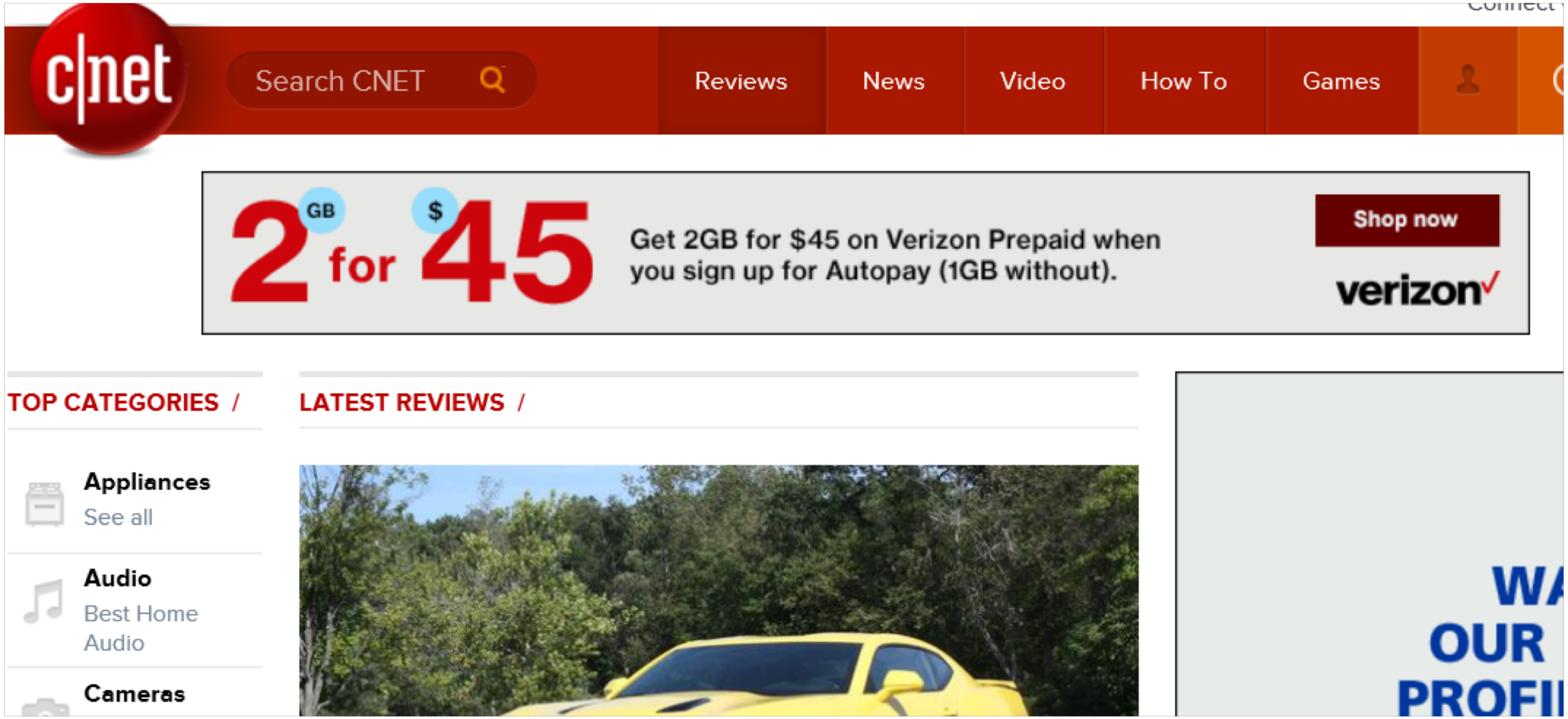
USC Viterbi School of Engineering
Information Sciences Institute

eBay: Direct Placement of Ads



USC Viterbi School of Engineering
Information Sciences Institute

Display Ads



c|net
Search CNET
Reviews
News
Video
How To
Games
2 GB for $ 45
Get 2GB for $45 on Verizon Prepaid when you sign up for Autopay (1GB without).
Shop now
verizon
TOP CATEGORIES /
LATEST REVIEWS /
Appliances
See all
Audio
Best Home Audio
Cameras

Online Algorithms

- **Classic model of algorithms**
  - You get to see the entire input, then compute some function of it
  - In this context, “offline algorithm”

- **Online Algorithms**
  - You get to see the input one piece at a time, and need to make irrevocable decisions along the way
  - **Make decisions without knowing the future**
  - **For search: only know past queries and current query; don’t know what queries will come in later**
  - Similar to handling data streams

- **An online algorithm cannot always do as well as an offline algorithm**

Example 8.1

- **Knowing the future could help**
- Manufacturer A of antique furniture bids 10 cents on search term "chesterfield"
- Manufacturer B of conventional furniture bids 20 cents on both terms "sofa" and "chesterfield"
- Both have monthly budget of $100
  - A can place its ad 1000 times, B can place its ad 500 times
- Query "chesterfield" arrives
- Can only display one ad
- Might display B's ad because B bid more, but…

Example 8.1

- **Knowing the future could help**
- Manufacturer A of antique furniture bids 10 cents on search term “chesterfield”
- Manufacturer B of conventional furniture bids 20 cents on both terms “sofa” and “chesterfield”
- Both have monthly budget of $100
  - A can place its ad 1000 times, B can place its ad 500 times
- Query “chesterfield” arrives
- Can only display one ad
- Might display B’s ad because B bid more
- However, if there are many queries for “sofa” and few for “chesterfield,” A will never spend its full budget
- Sending “chesterfield” queries to A might increase overall revenue
- **Without knowing the future, on-line algorithm may not perform as well as offline**

Offline Query-Ad Matching Problem

- Advertisers, each
  - Bids on keywords : “sofa”: 10 cents/impression
  - Has a budget, e.g., $100/mon
- A set of queries in some month, say Sep 2015
  - e.g., 600 “chesterfield”, 100 “sofa”
- Find assignments of queries to bids, such that
  - Total profit is maximized

Greedy Approach

- Consider two furniture manufacturers A and B
  - A: bids 20 cents on “chesterfield”; 10 cents on “sofa”
  - B: bids 10 cents on “chesterfield”
  - Both A and B have budget: $100/mon

- Queries: 600 “chesterfield”, 100 “sofa”
  - “chesterfield”: 500 to A => profit: $100
  - “chesterfield”: 100 to B => profit: $10

  => Total profit: $110

Non-Greedy Approach

- Consider two furniture manufacturers A and B
  - A: bids 20 cents on “chesterfield”; 10 cents on “sofa”
  - B: bids 10 cents on “chesterfield”
  - Both A and B have budget: $100/mon

- Queries: 600 “chesterfield”, 100 “sofa”
  - “sofa”: 100 to A => profit: $10
  - “chesterfield”: 450 to A => profit: $90
  - “chesterfield”: 150 to B => profit: $15

  => Total profit: $115

Optimal Solution

- Consider two furniture manufacturers A and B
  - A: bids 20 cents on “chesterfield”; 10 cents on “sofa”
  - B: bids 10 cents on “chesterfield”
  - Both A and B have budget: $100/mon

- Queries: 600 “chesterfield”, 100 “sofa”

- Optimal solution: assignment of queries to bids that generates the largest profit
  - e.g., non-greedy in previous slide is optimal

Comparison

Bids	Chestfield	Sofa	Budget
A	20 cents	10 cents	$100
B	10 cents		$100

Queries	Chestfield (600)	Sofa (100)	Profit
A	500		$100
B	100		$10

Greedy
Total profit: $110

Queries	Chestfield (600)	Sofa (100)	Profit
A	450	100	$100
B	150		$15

Non-Greedy
Total profit: $115

What if …?

- Consider two furniture manufacturers A and B
  - A: bids 20 cents on “chesterfield”; 10 cents on “sofa”
  - B: bids 10 cents on “chesterfield”
  - Both A and B have budget: $100/mon

- Query: 400 “chesterfield” and 100 “sofa”
  - Greedy approach works better => profit: $90
  - Any assignment of “chesterfield” to B will lose money, i.e., profit < $90

Online Bipartite Matching

The Matching Problem

- Simplified version of the problem of matching ads to search queries
- “Maximal matching”: involves bipartite graphs with two sets of nodes
- All edges connect node on left set to node in right set

Bipartite graph

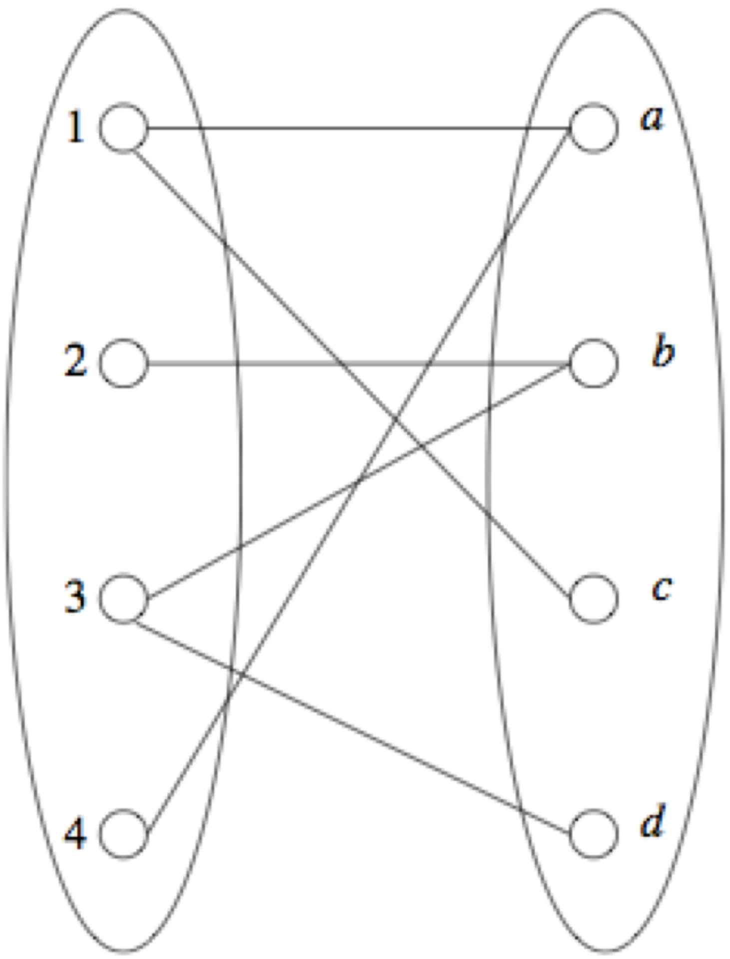
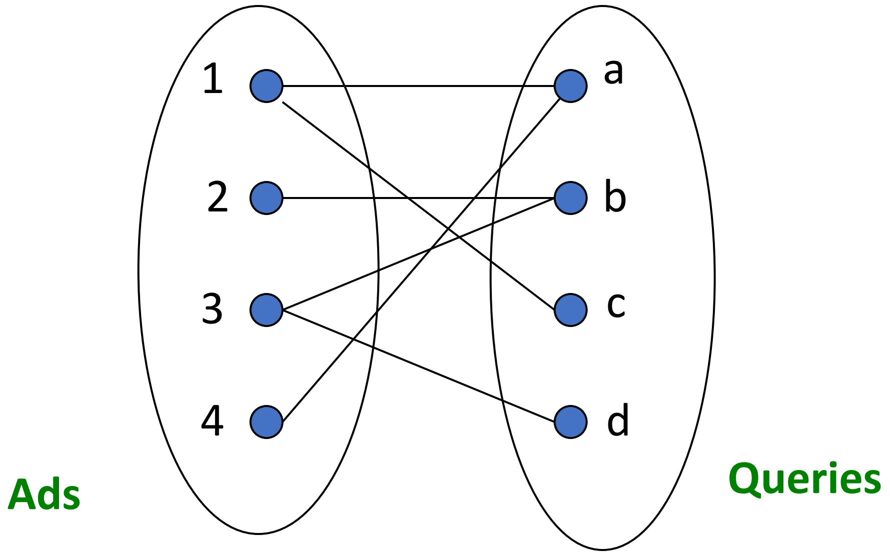


Figure 8.1: A bipartite graph

Example: Bipartite Matching

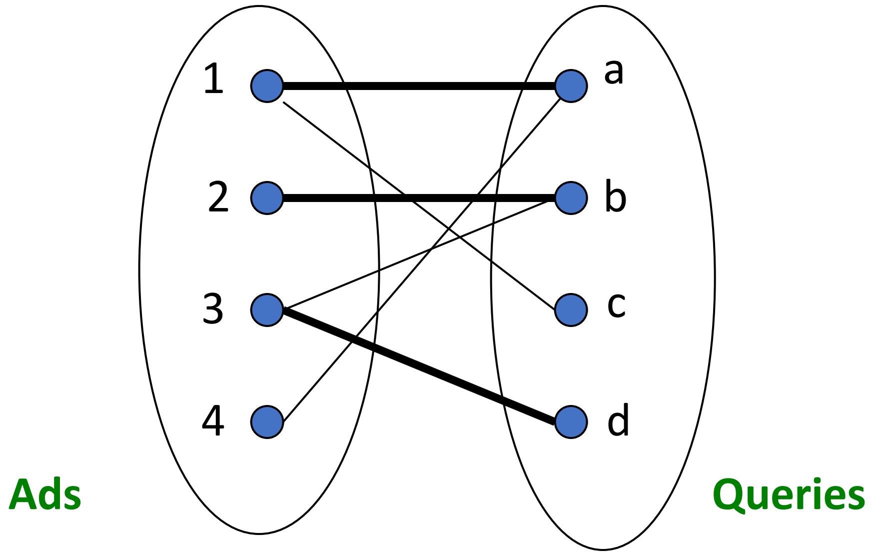


Nodes: Queries and Ads

Goal: Match queries to ads so that maximum number of matchings are made

USC Viterbi School of Engineering Information Sciences Institute

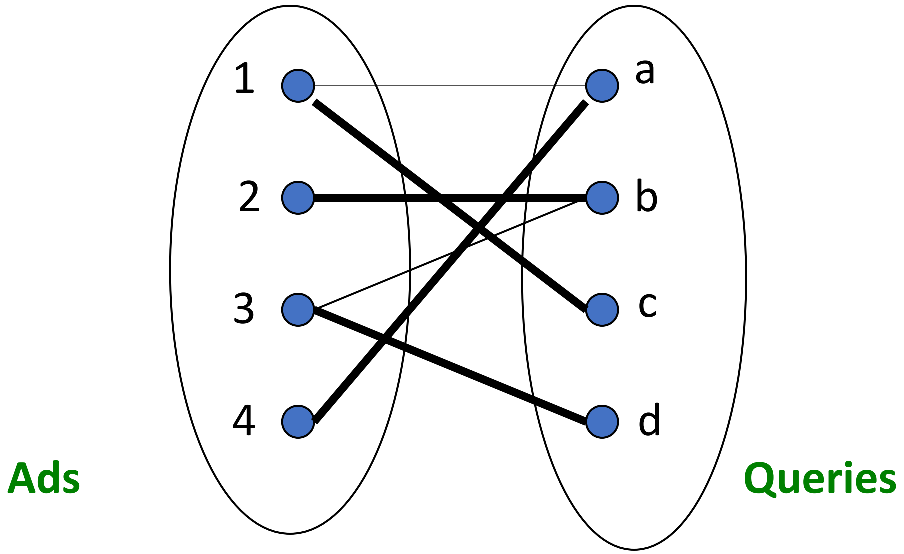
Example: Bipartite Matching



M = {(1,a),(2,b),(3,d)} is a **matching**

Cardinality of matching = |M| = 3

Example: Bipartite Matching



M = {(1,c),(2,b),(3,d),(4,a)} is a
perfect matching

Maximal matching: a matching that contains the largest possible number of matches
Perfect matching: all vertices of the graph are matched

Matching Algorithm

- **Problem: Find a maximal matching for a given bipartite graph**
  - A perfect one if it exists

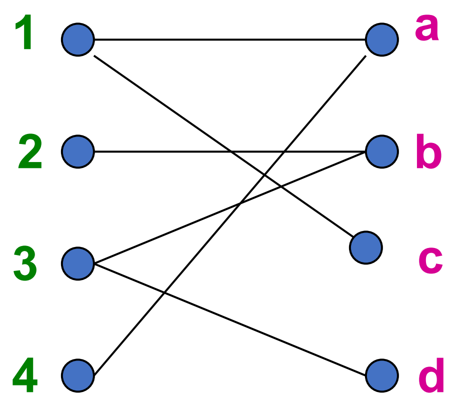
- There is a polynomial-time offline algorithm based on augmenting paths (Hopcroft & Karp 1973, see http://en.wikipedia.org/wiki/Hopcroft-Karp_algorithm)

- **But what if we do not know the entire graph upfront?**

Online Graph Matching Problem

- Initially, we are given the set **ads**
- In each **round**, **one set of query terms is added**
  - Relevant **edges** are revealed
  - Indicate which advertisers have bid on those query terms
- **At that time, we have to decide to either:**
  - Pair the **query** with an **ad**
  - Do not pair the **query** with any **ad**

Online Graph Matching: Example



(1,a)
(2,b)
(3,d)

Greedy Algorithm

- **Greedy algorithm for the online graph matching problem:**
  - Pair the new query with **any** eligible ad
    - If there is none, do not pair query

- **How good is the algorithm?**

Competitive Ratio

- For input ***I***, suppose greedy produces matching $\boldsymbol{M_{greedy}}$ while an optimal matching is $\boldsymbol{M_{opt}}$

Competitive ratio =

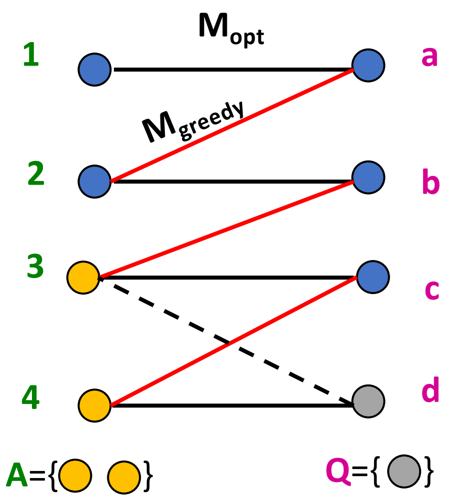
$$\min_{\text{all possible inputs } I} \left(|M_{greedy}| / |M_{opt}|\right)$$

(what is greedy's <u>worst</u> performance <u>over all possible</u> inputs *I*)

USC Viterbi School of Engineering Information Sciences Institute

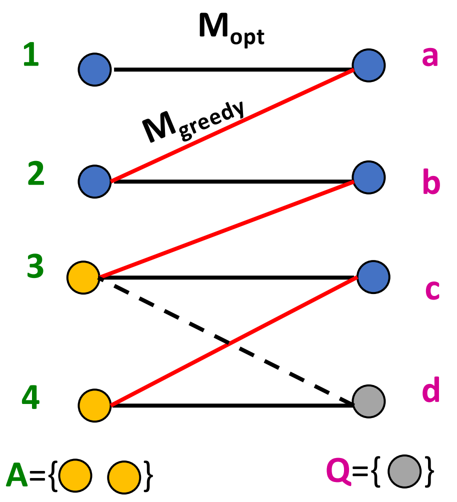
Analyzing the Greedy Algorithm

- Consider a case: $M_{greedy} \neq M_{opt}$
- Consider the set Q of queries matched in M_{opt} but not in M_{greedy}
- **A is the set of ads** that are adjacent (linked) to a non-matched query in Q that are already matched in M_{greedy}
  - If there exists such a non-matched (by M_{greedy}) ad adjacent to a non-matched query, then greedy would have matched them
- Since ads A are already matched in M_{greedy} then **(1)** $|M_{greedy}| \geq |A|$



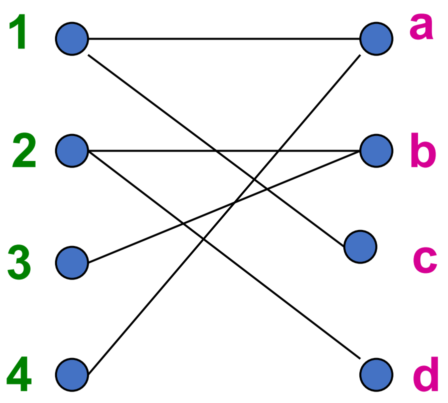
Analyzing the Greedy Algorithm

- **Summary so far:**
  - Queries Q matched in M_{opt} but not in M_{greedy}
  - (1) $|M_{greedy}| \geq |A|$
- There are at least $|Q|$ such ads in A ($|Q| \leq |A|$) otherwise the optimal algorithm couldn't have matched all queries in Q
  - **So:** $|Q| \leq |A| \leq |M_{greedy}|$
- Q': matched in M_{opt} and also in M_{greedy}
  - $| M_{opt} | = | Q | + | Q' |$ and $| Q' | \leq |M_{greedy}|$
- By definition of Q also: $|M_{opt}| \leq |M_{greedy}| + |Q|$
  - Worst case is when $|Q| = |A| = |M_{greedy}|$
- $|M_{opt}| \leq 2|M_{greedy}|$ **then** $|M_{greedy}|/|M_{opt}| \geq ½$



- **Competitive Ratio = ½**
- **Greedy's worst performance over all possible inputs *I***

Worst-case Scenario



(1,a)
(2,b)

- **Worst case** is when $|Q| = |A| = |M_{greedy}|$
- Q = {c,d} – queries with no matching ad
- A = {1,2} – ads that are adjacent to a query in Q but are already matched to another query
- $|M_{greedy}| = 2$, $|Q| = 2$, $|A| = 2$
- **Optimal matching: (1,c), (2,d), (3,b), (4,a)**
- $|M_{opt}| = 4$
- $|M_{greedy}|/|M_{opt}| = ½$ **(competitive ratio)**

Web Advertising

History of Web Advertising

- **Banner ads (1995-2001)**
  - Initial form of web advertising
  - Popular websites charged $X for every 1,000 "impressions" of the ad
    - Called "**CPM**" rate (Cost per thousand impressions)
    - Modeled similar to TV, magazine ads
  - From **untargeted** to **demographically targeted**
  - **Low click-through rates**
    - Low Return on Investment (ROI) for advertisers

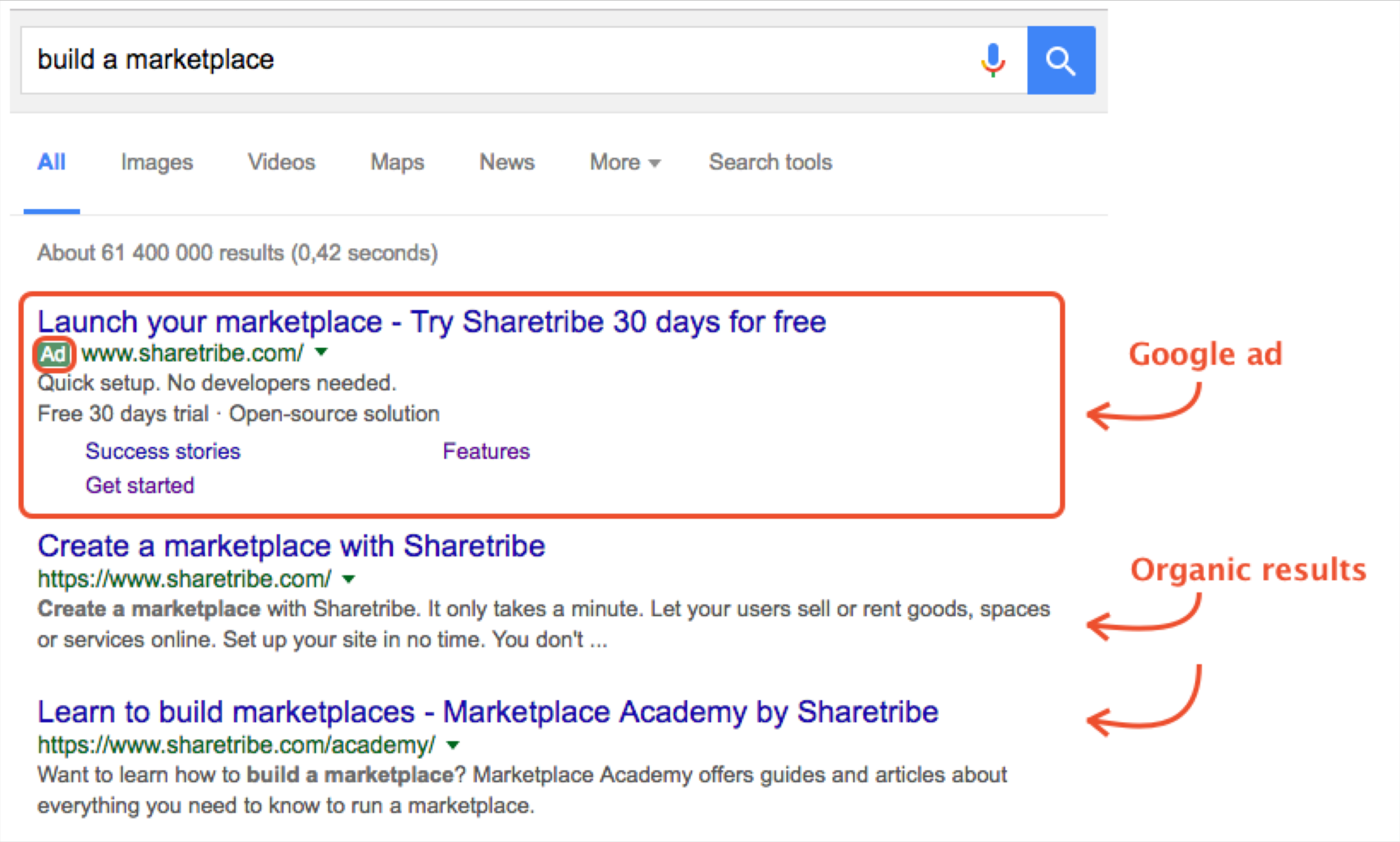
CPM…cost per *mille*
Mille…thousand in Latin

Performance-based Advertising

- **Introduced by Overture around 2000**
  - Advertisers **bid** on **search keywords**
  - When someone searches for that keyword, the **highest bidder's ad is shown first**
  - Advertiser is charged only if the ad is clicked on

- Similar model adopted by Google with some changes around 2002
  - Called **Adwords**

Ads vs. Search Results



Web 2.0

- **Performance-based advertising works!**
  - Multi-billion-dollar industry

- **Interesting problem:**
  What ads to show for a given query?
  - (Today's lecture)

- **If I am an advertiser, which search terms should I bid on and how much should I bid?**
  - (Not focus of today's lecture)

Adwords Problem

- **Given:**
  - **1.** A set of bids by advertisers for search queries
  - **2.** A click-through rate (CTR) for each advertiser-query pair
  - **3.** A budget for each advertiser (say for 1 month)
  - **4.** A limit on the number of ads to be displayed with each search query
- **Respond to each search query with a set of advertisers such that:**
  - **1.** The size of the set is no larger than the limit on the number of ads per query
  - **2.** Each advertiser has bid on the search query
  - **3.** Each advertiser has enough budget left to pay for the ad if it is clicked upon

Adwords Problem

- A stream of queries arrives at the search engine: q_1, q_2, …
- Several advertisers bid on each query
- When query q_i arrives, search engine must pick a subset of advertisers whose ads are shown

- **Goal: Maximize search engine's revenues**
  - **Simple solution:** Instead of raw bids, use the "**expected revenue per click**" (i.e., **Bid*CTR**)
- **Clearly we need an online algorithm!**

Invalid Clicks

- Thanks to:
  - Shweta Chandramouli
  - Siran Li
- More links:
  - http://www.google.com/ads/adtrafficquality/#click-fraud-and-invalid-traffic http://googleblog.blogspot.com/2008/03/using-data-to-help-prevent-fraud.html http://www.google.com/ads/adtrafficquality/invalid-click-protection.html
  - https://support.google.com/adwords/answer/2549113
  - https://support.google.com/adwords/answer/2375444
  - http://xcitemediagroup.com/competitors-clicking-my-ads-adwords-invalid-clicks
  - http://www.google.com/ads/adtrafficquality/index.html
  - https://support.google.com/adwords/answer/42995?hl=en

Adwords: Invalid Clicks

- https://support.google.com/adwords/answer/2375444#invalid

- **Invalid clicks** are basically clicks that Google doesn't consider "real" clicks, such as **clicks made by a robot or automated clicking tools**. You can see the number and percentage of clicks that have been classified as invalid and automatically filtered from your account by adding the "Invalid clicks" column on your **Campaigns** or **Dimensions** tabs. Don't worry, **you aren't charged for these clicks, and they don't affect your account statistics.**

More from Google…

- Here are just a few examples of what Google may consider to be invalid clicks:
  - **manual clicks intended to increase your advertising costs** or to **increase profits for website owners hosting your ads**
  - **clicks by automated clicking tools, robots, or other deceptive software**
  - **extraneous clicks that provide no value** to the advertiser, such as the second click of a double-click

More from Google…

- **Each click on an AdWords ad is examined by our system**, and Google has sophisticated systems to identify invalid clicks and impressions and remove them from your account data.

- When Google determines that clicks are invalid, we try to **automatically filter them from your reports and payments** so that you're not charged for those clicks. If we find that invalid clicks have escaped automatic detection, you may be eligible to receive a credit for those clicks. These credits are called "invalid activity" adjustments.

Using data to help prevent fraud

- http://googleblog.blogspot.com/2008/03/using-data-to-help-prevent-fraud.html
- Our Ad Traffic Quality team: three-stage system for detecting invalid clicks.
- The three stages are: **(1) proactive real-time filters**, **(2) proactive offline analysis**, and **(3) reactive investigations**.
- **Logs** provide us with the repository of data which are used to **detect patterns, anomalous behavior, and other signals indicative of click fraud.**
- filters (stage 1), which operate in real-time
- stages 2 and 3 which on deeper analysis of the data in our logs
- stage 2: pores over millions of impressions and clicks over longer time period
- **looking for unusual behavior** in hundreds of different data points
- **IP addresses:** for a given publisher or advertiser, where are their clicks coming from? Are they all coming from one country or city? Is that normal for an ad of this type?
- Don't identify individuals: **look at these in aggregate and study patterns**
- **Abnormally high number of clicks on a single publisher from the same ISP:** does look suspicious and raises a flag for us to investigate

The Adwords Innovation

Advertiser	Bid	CTR	Bid * CTR
A	$1.00	1%	1 cent
B	$0.75	2%	1.5 cents
C	$0.50	2.5%	1.125 cents

Click through rate

Expected revenue

The Adwords Innovation

Advertiser	Bid	CTR	Bid * CTR
B	$0.75	2%	1.5 cents
C	$0.50	2.5%	1.125 cents
A	$1.00	1%	1 cent

Complications: Budget

- **Two complications:**
  - **Budget**
  - **Click-through rate (CTR) of an ad is unknown**

- **Each advertiser has a limited budget**
  - **Search engine guarantees that the advertiser will not be charged more than their daily or monthly budget**

Complications: CTR

- **CTR: Each ad has a different likelihood of being clicked**
  - **Advertiser 1** bids \$2, click probability = 0.1
  - **Advertiser 2** bids \$1, click probability = 0.5
  - **Click-through rate (CTR)** is measured **historically**
    - **Very hard problem: Exploration vs. exploitation**
      Exploit: Should we keep showing an ad for which we have
      good estimates of click-through rate
      or
      Explore: Shall we show a brand new ad to get a better sense of its click-through rate

Greedy Algorithm

- **Our setting: Simplified environment**
  - There is **1** ad shown for each query
  - All advertisers have the same budget ***B***
  - All ads are equally likely to be clicked
  - Value of each ad is the same (=**1**)

- **Simplest algorithm is greedy:**
  - For a query pick any advertiser who has bid **1** for that query
  - **Competitive ratio of greedy is 1/2**

Bad Scenario for Greedy

- **Two advertisers A and B**
  - ***A*** bids on query ***x***, ***B*** bids on ***x*** and ***y***
  - Both have budgets of **$4**
- **Query stream: *x x x x y y y y***
  - Worst case greedy choice: ***B B B B _ _ _ _***
  - Optimal: **A A A A B B B B**
  - **Competitive ratio = ½**
- **This is the worst case!**
  - **Note:** Greedy algorithm is deterministic – it always resolves draws in the same way

Greedy algorithm with non-equal bids

- Greedy algorithm would **assign the query to the highest bidder who still has budget left**

Greedy Example:
Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1^{st} query q			
2^{nd} query q			
3^{rd} query q			
4^{th} query q			
5^{th} query q			
6^{th} query q			
7^{th} query q			
8^{th} query q			

Greedy Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A1	20	50
2nd query q			
3rd query q			
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

USC Viterbi School of Engineering Information Sciences Institute

Greedy Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A1	20	50
2nd query q	A1	0	50
3rd query q			
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

USC Viterbi School of Engineering Information Sciences Institute

Greedy Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A1	20	50
2nd query q	A1	0	50
3rd query q	A2	0	40
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

Greedy Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A1	20	50
2nd query q	A1	0	50
3rd query q	A2	0	40
4th query q	A2	0	30
5th query q	A2	0	20
6th query q	A2	0	10
7th query q	A2	0	0
8th query q	No ad	0	0

BALANCE Algorithm [MSVV]

- **BALANCE** Algorithm by Mehta, Saberi, Vazirani, and Vazirani
  - **For each query, pick the advertiser with the largest unspent budget**
    - Break ties arbitrarily (**but in a deterministic way**)

Example: BALANCE

- **Two advertisers A and B**
  - **A** bids on query ***x***, **B** bids on ***x*** and ***y***
  - Both have budgets of **$4**

- **Query stream:** ***x x x x y y y y***

- **BALANCE choice:** **A B A B B B _ _**
  - Optimal: **A A A A B B B B**

- **In general:** For **BALANCE** on **2** advertisers **Competitive ratio = ¾**

BALANCE Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q			
2nd query q			
3rd query q			
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

USC Viterbi School of Engineering Information Sciences Institute

BALANCE Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid x_1 = 20** **budget b_1 = 40**
- **Bidder A_2:** **bid x_2 = 10** **budget b_2 = 50**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q			
3rd query q			
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

BALANCE Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1^{st} query q	A2	40	40
2^{nd} query q	A1	20	40
3^{rd} query q			
4^{th} query q			
5^{th} query q			
6^{th} query q			
7^{th} query q			
8^{th} query q			

BALANCE Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q	A1	20	40
3rd query q	A2	20	30
4th query q			
5th query q			
6th query q			
7th query q			
8th query q			

USC Viterbi School of Engineering Information Sciences Institute

BALANCE Example:
Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q	A1	20	40
3rd query q	A2	20	30
4th query q	A2	20	20
5th query q			
6th query q			
7th query q			
8th query q			

USC Viterbi School of Engineering Information Sciences Institute

BALANCE Example:
Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q	A1	20	40
3rd query q	A2	20	30
4th query q	A2	20	20
5th query q	A1	0	20
6th query q			
7th query q			
8th query q			

USC Viterbi School of Engineering Information Sciences Institute

BALANCE Example: Two advertisers bid on a query q

- **Bidder A_1:** **bid x_1 = 20** **budget b_1 = 40**
- **Bidder A_2:** **bid x_2 = 10** **budget b_2 = 50**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q	A1	20	40
3rd query q	A2	20	30
4th query q	A2	20	20
5th query q	A1	0	20
6th query q	A2	0	10
7th query q			
8th query q			

BALANCE Example:
Two advertisers bid on a query q

- **Bidder A_1:** **bid $x_1 = 20$** **budget $b_1 = 40$**
- **Bidder A_2:** **bid $x_2 = 10$** **budget $b_2 = 50$**
- Assume ties are broken in favor of A_1

Query q	Assigned to Bidder (A_1, A_2 or No Ad)	Remaining Budget for A_1	Remaining Budget for A_2
At start	----	40	50
1st query q	A2	40	40
2nd query q	A1	20	40
3rd query q	A2	20	30
4th query q	A2	20	20
5th query q	A1	0	20
6th query q	A2	0	10
7th query q	A2	0	0
8th query q	No Ad	0	0

Analyzing BALANCE

- **Consider simple case (w.l.o.g.):**
  - **2** advertisers, $\mathbf{A_1}$ and $\mathbf{A_2}$, each with budget **B** ($\geq$1)
  - Optimal solution exhausts both advertisers' budgets

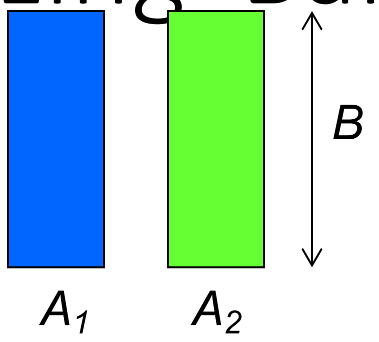
- **BALANCE must exhaust at least one advertiser's budget:**
  - **If not, we can allocate more queries**
    - Whenever both advertisers bid on the query, chosen advertiser's unspent budget only decreases
    - In BALANCE, one budget will be exhausted

- Assume BALANCE exhausts A_2's budget, but allocates x queries fewer than the optimal

- **Revenue:** $BAL = 2B - x$

USC Viterbi School of Engineering *Information Sciences Institute*

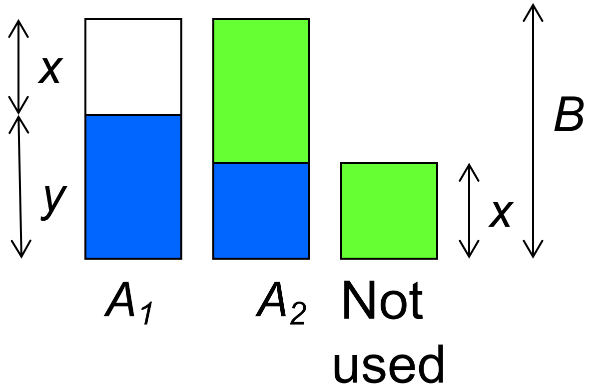
Analyzing Balance



■ Queries allocated to A_1 in the optimal solution

■ Queries allocated to A_2 in the optimal solution

Optimal revenue = **2B**



Balance Algorithm:

Assume Balance gives revenue **= 2B-x or B+y**

Unassigned queries should be assigned to A_2

(if we could assign to A_1 we would, since we still have budget)

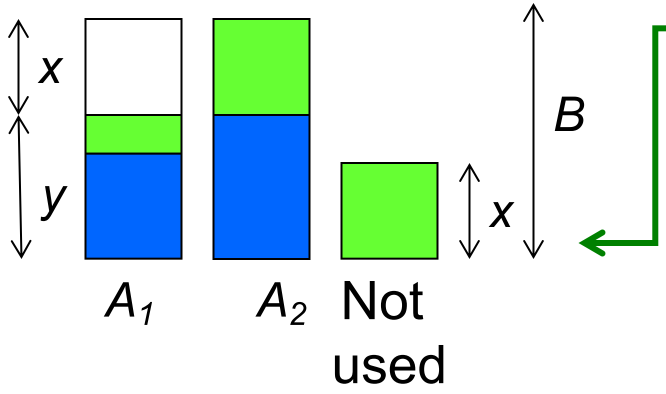
Goal: Show we have $y \geq x$

Case 1) ≤ ½ of A_1's queries got assigned to A_2

then y >= B/2, so surely y >= x (y+x = B)

Case 2) > ½ of A_1's queries got assigned to A_2

then x < B/2 and x + y = B so y >= x



Balance revenue is minimum for x=y=B/2

Minimum Balance revenue = 3B/2

Competitive Ratio = 3/4

BALANCE exhausts A_2's budget

BALANCE: General Result

- **For Balance algorithm with many bidders**
- **In the general case, worst competitive ratio of BALANCE is 1–1/e = approx. 0.63**
  - Interestingly, no online algorithm has a better competitive ratio!

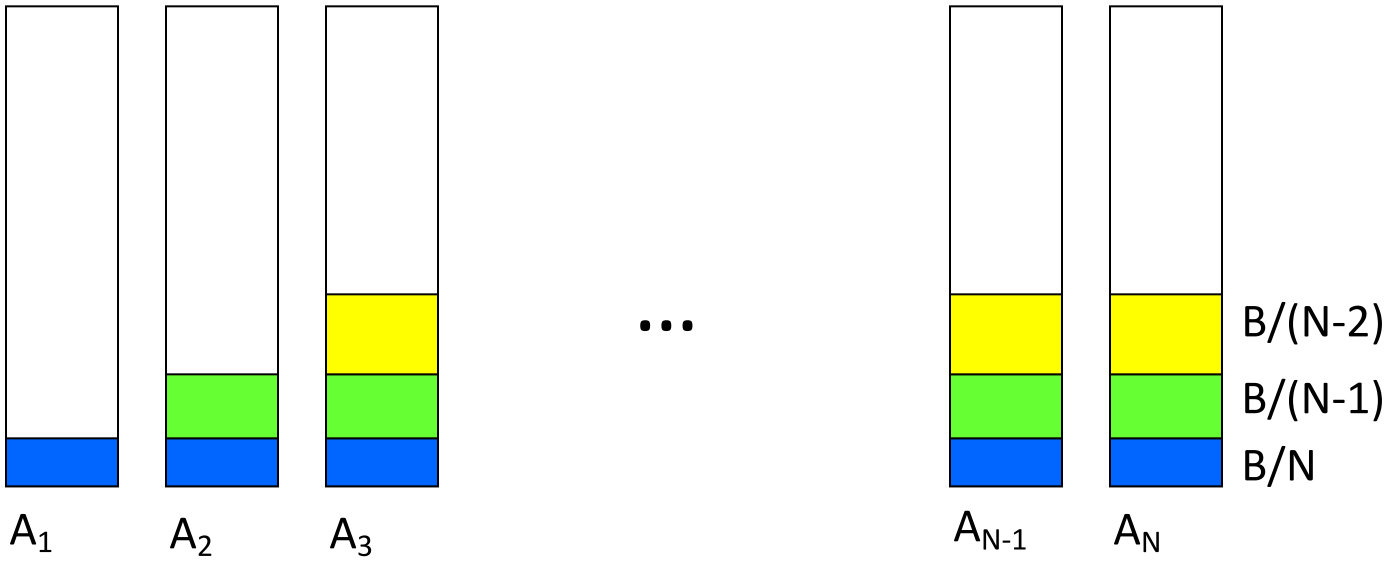
- **Let's see the worst case example that gives this ratio**

Worst case for BALANCE

- ***N* advertisers: $A_1, A_2, \dots A_N$**
  - Each with budget $B = N$
- **Queries:**
  - $N \cdot B$ queries appear in N rounds of B queries each
- **Bidding:**
  - Round **1** queries: bidders $A_1, A_2, \dots, A_N$
  - Round **2** queries: bidders $A_2, A_3, \dots, A_N$
  - Round ***i*** queries: bidders $A_i, \dots, A_N$
- **Optimum allocation:**
  Allocate round ***i*** queries to A_i
  - Optimum revenue $N \cdot B$
- **BALANCE:**
  - Assigns each query in round 1 to N advertisers equally, since all bid on q1
  - Prefers bidder with largest remaining budget
  - For q2, divided equally among $A_2, A_3, \dots, A_N$
  - For each query qi, advertisers $A_i, \dots, A_N$ get queries

USC Viterbi School of Engineering Information Sciences Institute

BALANCE Allocation



- **Eventually, budgets of higher-numbered advertisers exhausted**
- **Want j such that ln N - ln (N-j) = 1 (approximately)**

$$B\left(\frac{1}{N}+\frac{1}{N-1}+\cdots+\frac{1}{N-j+1}\right) \geq B$$

Euler showed that as k gets large, $\sum_{i=1}^{k} 1/i$ approaches $\log_e k$.

$$\log_e N - \log_e(N-j) = 1,$$

- j = N(1-1/e) is approximate value where all advertisers are out of budget or do not bid on remaining queries
- **Approximate revenue of Balance Algorithm is BN(1-1/e)**
- **Competitive ration is 1-1/e**

General Version of the Problem

- **Balance works well when bids are 1,0**
- **In practice, bids and budgets can be arbitrary**
- Consider we have 1 query ***q***, advertiser ***I***
- **In a general setting, BALANCE can perform poorly**
- **Example 8.9:** Consider two advertisers A_1 and A_2
  - A_1: bid_1 = **1**, $budget_1$ = **110**
  - A_2: bid_2 = **10**, $budget_2$ = **100**
  - Consider: we see **10** instances of **q**
  - BALANCE always selects A_1 because it has largest remaining budget
  - Earns total revenue = **10**
  - **Favors advertiser with larger remaining budget**
  - Optimal earns **100**

USC Viterbi School of Engineering Information Sciences Institute

Modifications Needed to BALANCE Algorithm

- **Bias choice of ad in favor of higher bids**
- **Consider the fraction of budget remaining,** so we **bias toward using some of each advertiser's budget**
- More "risk averse": don't leave too much of any advertiser's budget unused

Generalized BALANCE Algorithm

- **Arbitrary bids:** consider query $\boldsymbol{q}$, bidder $\boldsymbol{i}$
  - Bid = $\boldsymbol{x_i}$
  - Budget = $\boldsymbol{b_i}$
  - Amount spent so far = $\boldsymbol{m_i}$
  - **Fraction of budget left over $f_i = 1- (m_i/b_i)$**
- Define $\psi_i(q) = x_i * (1-e^{-f_i})$ ψ (psi)
  - $bid * (1-e^{-(\text{fraction of budget left})})$
- Allocate query $\boldsymbol{q}$ to bidder $\boldsymbol{i}$ with largest value of $\psi_i(q)$
- **Same competitive ratio (1-1/e)**

USC Viterbi School of Engineering
Information Sciences Institute

Example 8.10

- **Bidder A_1: $x_1 = 1$, $b_1 = 110$**
- **Bidder A_2: $x_2 = 10$, $b_2 = 100$**
- First occurrence of query q: **fraction 1 of budgets b_1 and b_2 remain**
- ***$\psi_1(q) = x_1(1-e^{-f_1}) = 1(1-e^{-1}) = 1 - 1/e = 0.63$***
- ***$\psi_2(q) = x_2(1-e^{-f_2}) = 10(1-e^{-1}) = 6.3$***
- **So first q is awarded to A_2**
- ***$\psi_2(q)$* decreases, but for the next 9 instances of q: *$\psi_2(q) > \psi_1(q)$* and queries are awarded to A_2**
- For 10^{th} instance of q, **remaining fraction of budget b_2 is 1/10**
- ***$\psi_2(q) = x_2(1-e^{-f_2}) = 10(1-e^{-1/10}) = 0.95$, which is > 0.63***
- **After 10 queries q, have spent all of A_2's budget, and additional queries q will be awarded to A_1**
- **Total revenue for 10 queries q = 100**
- **Generalized Balance Algorithm: Successfully biased toward higher bids, took into account fraction of budget remaining**

Additional Observations

- Algorithm as described **does not account for possibility that click-through rate differs for different ads**
- **Multiply bid by CTR when computing ψ**

- Also can **consider historical frequency of queries**
  - **Use historical frequency to predict future frequency**

Adwords Aspects Not in Our Model

Matching bids and search queries:

- **In our simplified model, advertisers bid on sets of words**
- An advertiser's bid is eligible to be shown for search queries with **exactly the same set of words as advertiser's bid**

- In reality, **Google, Yahoo, Microsoft all offer advertisers "broad matching": inexact matches of the bid keywords**
- **Examples: subsets, supersets, words with very similar meanings**

- Charge advertisers based on **complicated formulas that take into account how closely related the search query is to the advertiser's bids**
- **Proprietary algorithms**

Adwords Aspects Not in Our Model

Charging Advertisers for Clicks

- **In our simplified model, when a user clicks on an ad, the advertiser is charged the amount they bid**
- Known as a **first-price auction**
- **In reality, search engines use a more complicated system known as a second-price auction**
- **Each advertiser pays approximately the bid of the advertiser who placed immediately behind them in the auction**
  - Example: **First-place advertiser would pay the bid of the second-place advertiser plus one cent**
- **Less susceptible to being gamed by advertisers than first-price auctions**
- **Lead to higher revenues for search engines**
  - https://blogs.cornell.edu/info2040/2012/10/27/google-adwords-auction-a-second-price-sealed-bid-auction/