Analysis of Massive Data Sets

http://www.fer.hr/predmet/avsp

Prof. dr. sc. Siniša Srbljić

Doc. dr. sc. Dejan Škvorc

Doc. dr. sc. Ante Đerek

Faculty of Electrical Engineering and Computing
Consumer Computing Laboratory

Advertising on the Web

Goran Delač, PhD

Outline

- **Motivation**
  - Advertising on the Web, Issues
- **Online algorithms**
  - Online Bipartite Matching
  - Greedy algorithm
- **Advertising on the Web**
  - Adwords problem
  - Solutions: Greedy algorithm, Balance algorithm

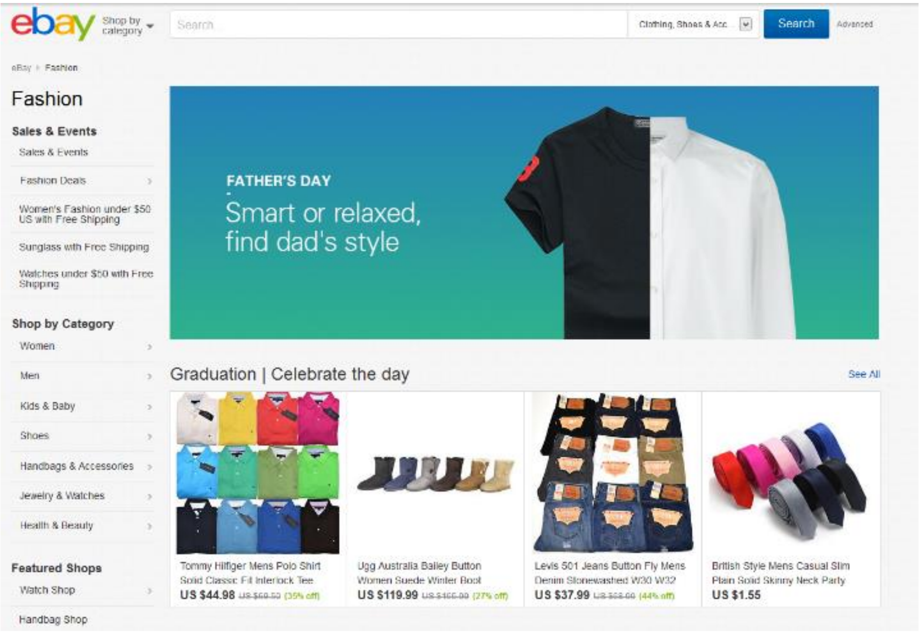
Motivation

- **Web applications support themselves through advertising**
  - One of the big surprises of the 21st century
  - Multi-billion dollar market

- **Other media revenue sources**
  - Radio and television
    - Advertising (primary)
  - Most media (newspapers and magazines)
    - Subscription (primary)
    - Hybrid approach – advertising and subscriptions

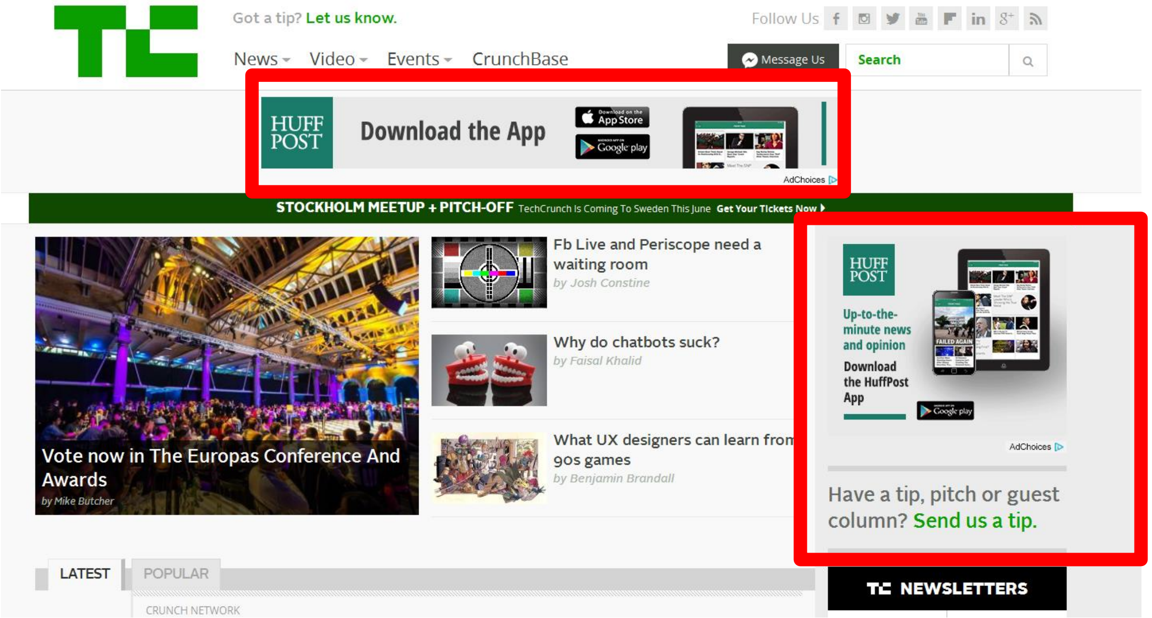
Motivation: Ad Types

- **Direct Placement of Ads**
  - Directly placed by advertisers
    - Free, for a fee or commission
  - eBay, Craig's List



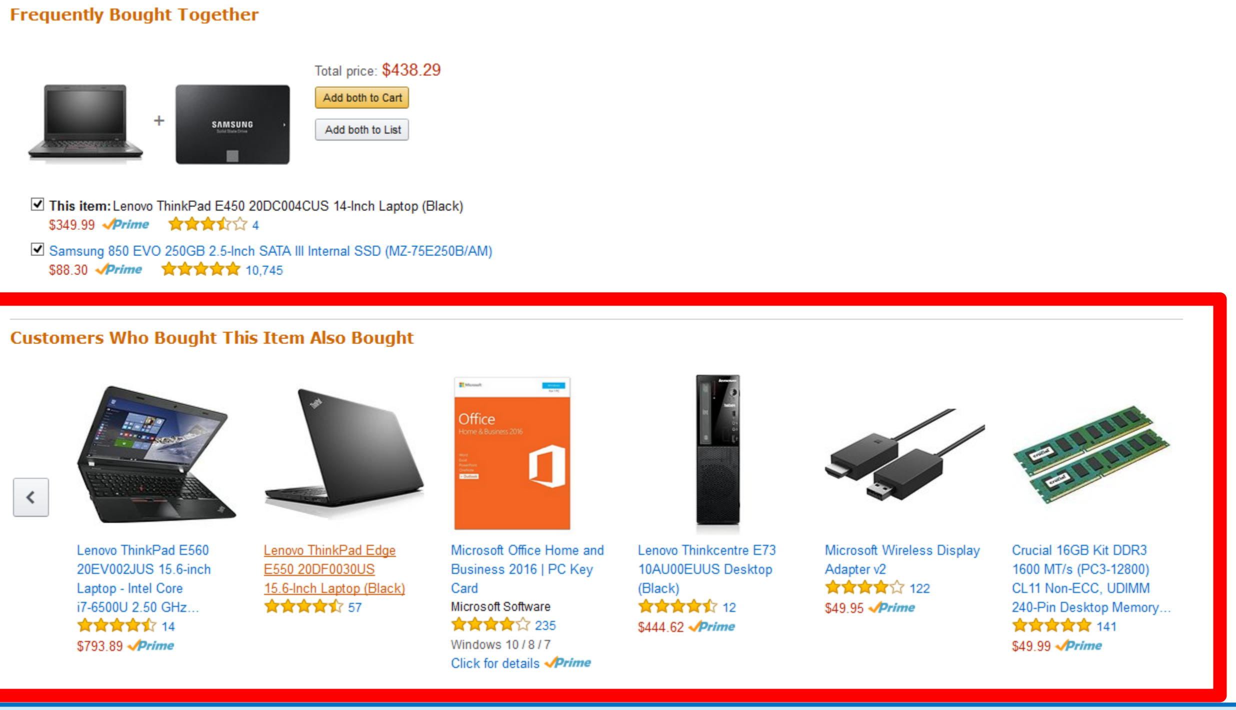
Motivation: Ad Types

- **Display ads (banners)**
  - Earliest form of Web advertising
  - E.g. fixed rate for *impression* (total cost defined by the number of times an ad has been rendered in a browser)



Motivation: Ad Types

- **Recommendation systems**
  - Amazon
  - Ad is selected by the store to maximize the probability that a customer will be interested in a product



Motivation: Ad Types

- **Search ads**
  - Placed along with the results of a search query
  - Bids for the right to have ad shown in response to certain queries
    - Payed only if the ad is clicked on

Motivation: Issues

- **Direct Placement of Ads**
  - **Identifying ads:** displayed in response to query terms
    - Inverted index of words (search engine)
    - Filters (user specifies a set of predefined parameters)
  - **Assigning importance to ads:** display order
    - Most recent first (njuskalo.hr)
      - Display the latest ad
      - Possible abuse – minimal ad changes at frequent intervals
    - Measure the “**attractiveness**” of an add
      - Record how many times the ad has been clicked on
      - Ads that are clicked on more frequently are presumed to be more attractive

Motivation: Issues

- **Direct Placement of Ads**
  - Measuring ad “**attractiveness**” is not that straightforward
    - The position of an ad in a list: first ad in a list has by far the greatest probability to be clicked on
    - Attractiveness can depend on the query terms
    - All ads should have opportunity to be shown initially (until their click probability can be estimated their attractiveness is unknown)

Motivation: Issues

- **Display Ads**
  - Resemble advertising in traditional media
  - Problem: lack of focus
    - User might not be interested (e.g. just bought a new mobile phone or is generally not into tech)
    - Low click through rates (banner advertising 1995-2001 offered low return of investment)
  - Printed media / TV cannot solve this issue **but** web ads can!

Motivation: Issues

- **Display Ads**
  - User-tailored ads: adds fitted to a specific user
    - use information about users to determine which ad they should be shown
  - **Problem:** need to get to know users
    - Activity on Facebook
    - E-mail
    - Time spent on a particular site, bookmarks etc.
    - **Search queries**
  - Significant privacy issues

Motivation: PBA

- Performance-based Advertising

- Concept introduced by Overture (2000)
  - Advertisers place bids on search keywords
  - When a keyword is searched, the **highest bidder's ad is shown**
  - Advertiser is charged **only if** the ad is clicked on

Motivation: PBA

- **Google adopted the Overture PBA model around 2002 and modified it**
  - **Adwords**
  - The value of PBA was proven as web advertising started to get a lot of traction

Motivation: PBA

- **What ads are to be displayed for a given user query?**

- **What search terms advertisers should bid on?**
- **How much to bid for a particular search term?**
  - Out of scope of this lecture

Motivation: PBA

- **Problem**
  - Advertisers usually have a limited budget
  - How to display ads in an optimal way if all search queries are **not known** in advanced?
  - **Online algorithms**

Online algorithms

- **Classic (“offline”) algorithms**
  - The entire input is available
  - Can access data set in any order and compute some function over all input values

- **Online Algorithms**
  - Do not have access the entire data set
  - Input is read piece-by-piece
  - Output is produced for each input data set piece (cannot wait for the entire data set to arrive!)

Online algorithms

- **Examples:**
  - Stream mining algorithms
  - Insertion sort
  - BFR clustering algorithm (if centroids are known)
  - …

- **Many online algorithms are greedy**
  - Output is produced by maximizing some function of the current input and the past (comprising of previous input elements)

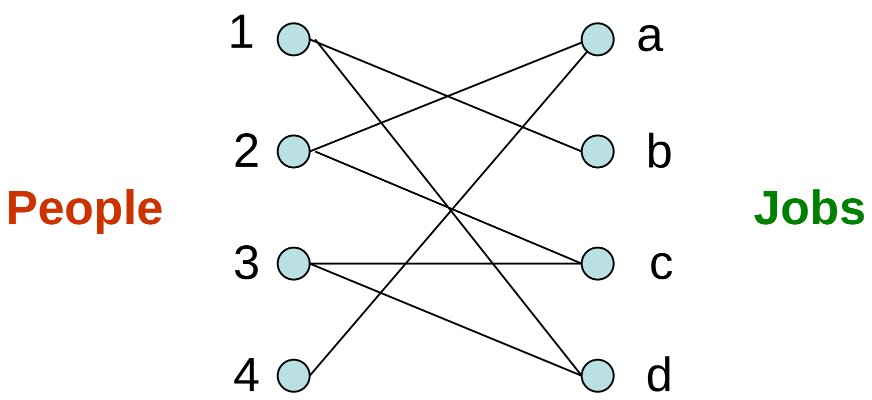
Online algorithms

- **Online algorithms can (and usually do) return result that is not as good as the result of the best offline algorithm**

- **Competitive ratio**
  - Given a solution quality for an offline algorithm $\boldsymbol{o}$ there exists such constant $\boldsymbol{c}$, $0 \leq \boldsymbol{c} \leq 1$, so that $\boldsymbol{c} \cdot \boldsymbol{o}$ is the solution quality of the online algorithm
  - $\boldsymbol{c}$ is the ***competitive ratio*** for the online algorithm

Online algorithms: Example

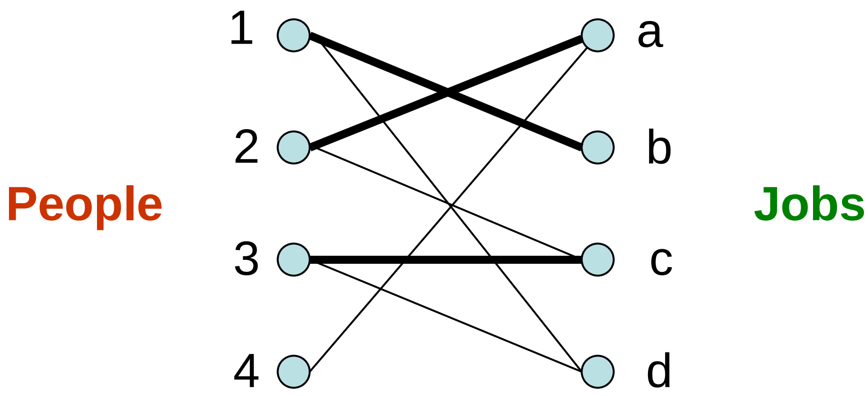
- **The Bipartite Matching Problem**
  - Assignment of entities from two ("different") sets
  - E.g. assign people to jobs, tasks to servers
  - OR **ads** to web site renderings!



  - **Constraints:** number of ways entities can be matched is limited → **defined** by graph edges
    - Limited number of job listings, job limitations, …

Online algorithms: Example

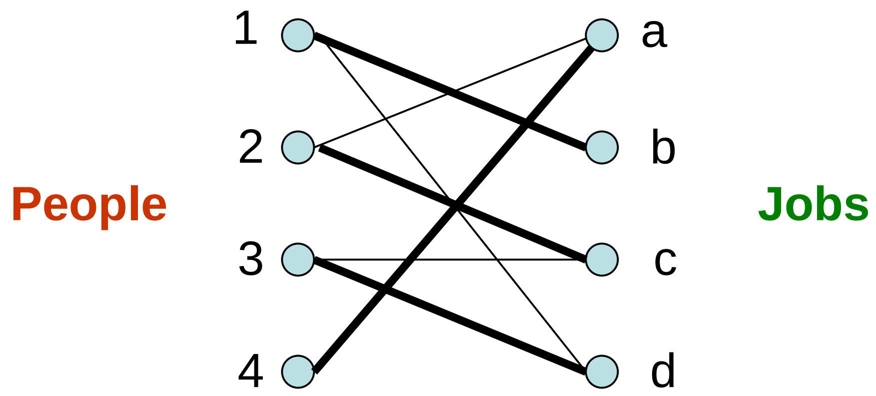
- **The Bipartite Matching Problem**
  - **Task:** prune the graph so each vertex in set s_1 is connected to **at most one** vertex in set s_2



  - Matched pairs (1,b), (2,a), (3,c)
  - Cardinality of matching $|\mathbf{M}| = \mathbf{3}$

Online algorithms: Example

- **The Bipartite Matching Problem**
  - **Maximum matching:** largest possible number of matches → **Goal**
  - **Perfect matching:** all graph vertices are matched



  - Perfect matching = maximum matching
  - More than one maximum/perfect matching can exist

Online algorithms: Example

- **The Bipartite Matching Problem**
  - **Offline algorithms** solve the problem of finding maximal matching in polynomial time
  - Hopcroft-Karp algorithm ($O\left(E\sqrt{V}\right)$)
  - Can be treated as a max-flow problem
  - What if entire data set is not known in advance?

Online algorithms: Example

- **Online approach to bipartite matching problem**
  - Initially, set s_1 is known (e.g. people that look for jobs, possible ads)
  - Data from set s_2 becomes known gradually (job offers, ad rendering possibilities)
  - Choice:
    - Pair available items using current knowledge
    - Do not pair items (a better match could become available)

Online algorithms: Greedy Approach

- **Matching pairs in a greedy manner**
  - Pair the newly discovered entity with any available (but eligible) entity in the other set
    - Pair the job offering with the first available person
    - If eligible person does not exist, do not match
  - Is this approach any good?

Online algorithms: Greedy Approach

- Let D be the input data set
- Let be M_G and M_0 the cardinality of matching for greedy and offline algorithms respectively

- Competitive ratio c is defined as:

$$c = \min_{D}\left(\frac{M_G}{M_0}\right)$$

  - c represents the **worst case** scenario for the greedy approach

Online algorithms: Greedy Approach

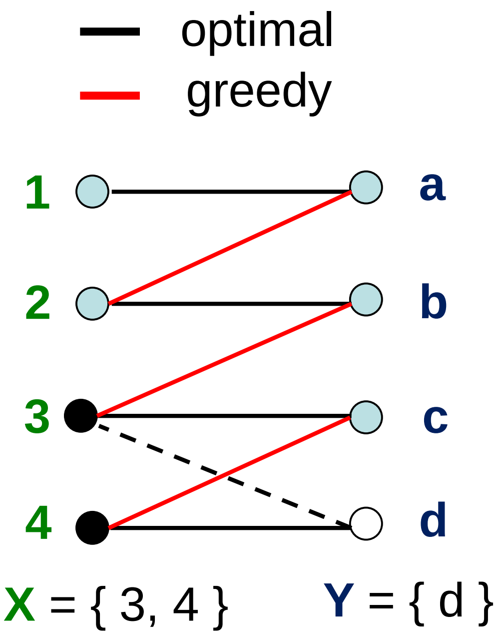
- **Let $M_G \neq M_O$**

- Y – a set of unmatched entries in M_G (e.g. jobs)
- X – a set of entities adjacent to Y

$$M_o = M_G + |Y| \qquad (1)$$

$$M_G \geq |X| \qquad (2)$$

As all elements in X are apparently matched by the greedy algorithm. Otherwise, elements in Y would have been matched!



Online algorithms: Greedy Approach

- **Let $M_G \neq M_O$**

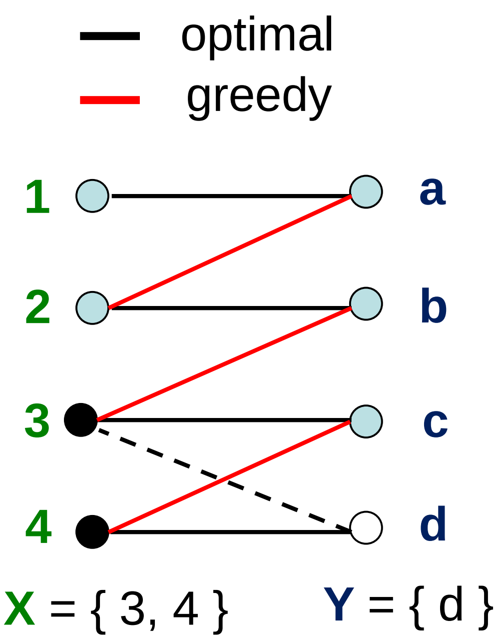
$$|Y| \leq |X| \qquad (3)$$

As optimal approach matched all elements in Y to some elements in X

$$(2) + (3) \rightarrow |Y| \leq |X| \leq M_G \qquad (4)$$

Worst case

$$|Y| = |X| = M_G \qquad (5)$$

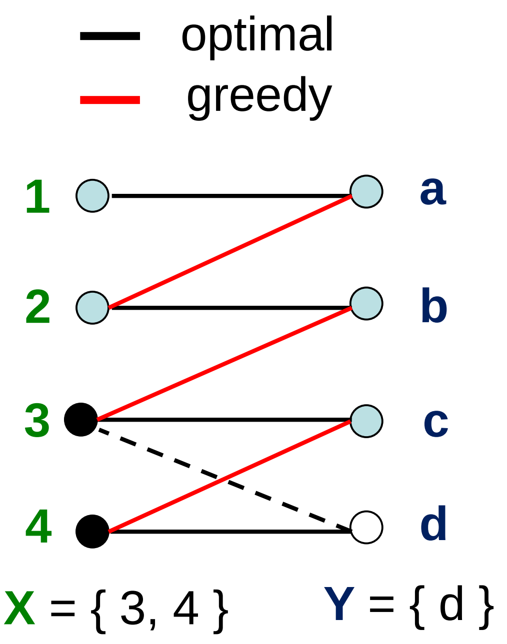


Online algorithms: Greedy Approach

- **Let $M_G \neq M_O$**

(1) + (5) → $M_O \leq 2M_G$

$$\boxed{\frac{M_G}{M_O} \geq 1/2}$$



Advertising: The Adwords Problem

- **How to match ads to search queries?**
  - Very similar to the general problem of bipartite graph matching

- **Search engine gets a set of queries as input values**
- **Advertisers bid on search keywords**
- **Upon answering the query, the search engine picks a subset of ads to display**
  - Usually more than one ad is shown

- **Goal is to maximize the profits from advertising**

Advertising: Adwords problem

Advertiser	Bid
A	$1.00
B	$0.75
C	$0.50
D	$0.25

Model used by Overture
Advertisers are sorted by bid values

Advertising: The Adwords Problem

- **Adwords introduced the click-through rate (CTR)**

- **CTR**
  - Number of times the ad has been clicked on as the result of being displayed divided by the total number of ad clicks

- **Expected revenue**
  - B * CTR, where B is the bidding value

Advertising: Adwords problem

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
D	$0.25	8%	2 cents

Advertising: Adwords problem

Advertisers sorted by the expected revenue

Advertiser	Bid	CTR	Bid * CTR
D	$0.25	8%	2 cents
B	$0.75	2%	1.5 cents
C	$0.50	2.5%	1.125 cents
A	$1.00	1%	1 cent

Advertising: Adwords problem

- **Statement of the Adwords problem**

- **Having the following values:**

  1. A set of advertisers' bids on search queries
  2. A click-through rate for each ad-query(keyword) pair
  3. A budget for each advertiser (e.g. for 1 month)
  4. A limit on the number of ads that can be displayed per search query

Advertising: Adwords problem

- **Derive a subset of ads such that:**

1. The size of the set is not larger than the ad display limit
2. The advertiser has placed a bid on the search keywords
3. The advertiser has enough budget left to pay if the ad gets clicked on

Advertising: Adwords problem

- **It turns out that sorting the ads by the expected revenue is the optimal strategy, but only if**
  - The click-trough rate for each ad-query pair is known
  - Advertisers have an unlimited budget
- **In general, this is not the case!**
- **How to estimate the CTR and deal with limited advertiser budgets?**

Advertising: Estimating CTR

- **CTR can be measured historically**
  - Show the ad a large number of times
  - Compute the CTR estimate by observing the number of clicks

- **Problems**
  - CTR is very position dependent
    - First ad in the list has the greatest probability to be clicked on
  - Explore or Exploit
    - Exploit: continue displaying ads with a high CTR estimate
    - Explore: examine CTR of a new ad (might be worthwhile)

Advertising: Limited Budget

- **Simplified environment**
  - **1** ad is shown for each query
  - All advertisers have the same budget ***B***
  - All ads are equally likely to be clicked
  - All ads have the same price (e.g. 1 cent)

- **Greedy algorithm:**
  - Pick an advertiser that has bid on the query and has enough leftover budget
  - Competitive ratio is 0.5

Advertising: Limited Budget

- **Two advertisers A and B**
  - ***A*** bids only on queries $\boldsymbol{x}$, ***B*** bids both on $\boldsymbol{x}$ and $\boldsymbol{y}$
  - Both advertisers have budgets of **4 cents**
  - Cost of displaying an ad is 1 cent

- **Query stream: *x x x x y y y y***
  - Worst case for greedy: ***B B B B*** _ _ _ _
    - Earned 4 cents
  - Optimal solution: **A A A A B B B B**
    - Earned 8 cents

Advertising: Limited Budget

- **Is it possible to get a better competitive ratio?**
  - Problem with greedy approach is that it always breaks ties in the same way
  - This leads to exhausting budgets of certain advertisers and thus limiting the ways of efficiently utilizing other advertisers

Advertising: BALANCE Algorithm

- **BALANCE Algorithm (Mehta, Saberi, Vazirani, and Vazirani)**
  - Upon processing a query, choose the advertiser with the largest unspent budget
  - Break ties in an arbitrary way
    - But do it **deterministically**

Advertising: BALANCE Algorithm

- **Two advertisers A and B**
  - ***A*** bids only on queries x, ***B*** bids both on x and y
  - Both advertisers have budgets of **4 cents**
  - Cost of displaying an ad is 1 cent

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

	Break tie		Break tie					
Balance A	4	3	3	2	2	2	2	2
Balance B	4	4	3	3	2	1	0	0
Output	A	B	A	B	B	B	-	-

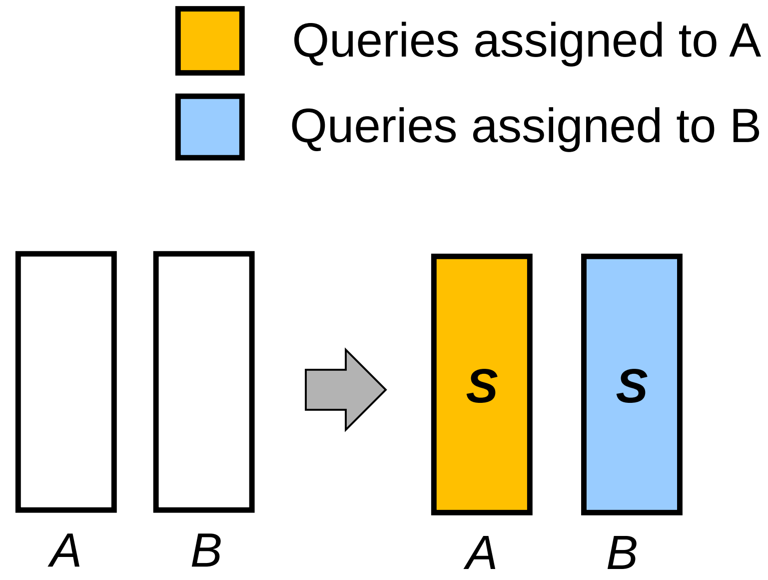
Advertising: BALANCE Algorithm

- **BALANCE output: A B A B B B _ _**
  - Optimal: A A A A B B B B
  - Cardinality of matching is ¾

- **In general: for BALANCE having 2 advertisers**
  - $\mathbf{Competitive\ ratio} = \frac{\mathbf{3}}{\mathbf{4}}$

Advertising: BALANCE Algorithm

- **Proof c = ¾**
  - Two advertisers A and B
  - They have the same budget S
  - Let all ads be priced the same (=1)
  - 2S queries
  - Optimal case
    - Both budgets get exhausted
    - The overall revenue is **2S**

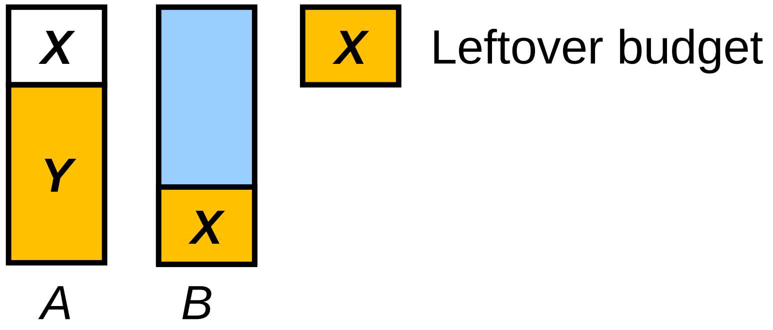


Advertising: BALANCE Algorithm

- **Proof c = ¾**
  - Budget in case of BALANCE algorithm
    - Budget of **one advertiser** will surely get **exhausted**!
      - Because optimal approach managed to perform a perfect match
      - Both advertisers placed a bid on at least half the queries
    - Assume without the loss of generality that B’s budget gets exhausted

Advertising: BALANCE Algorithm

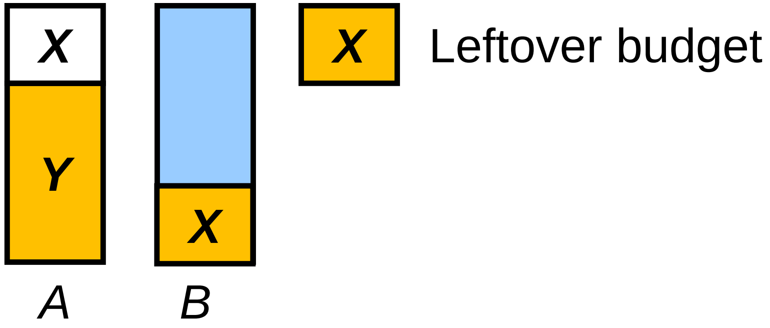
- **Proof c = ¾**
  - Budget in case of BALANCE algorithm
    - X – number of queries that were left unassigned
  - Total revenue in this case is
    - 2S – X
    - S + Y



Advertising: BALANCE Algorithm

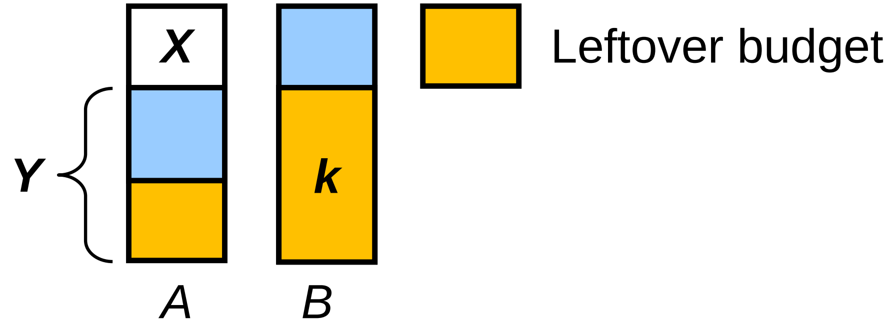
- **Proof $c = 3/4$**
  - Goal
    - Prove that $X \leq S/2$ or $Y \geq S/2$
    - Thus, $Y \geq X$
  - Case 1
    - Assume that less then S/2 queries of A were assigned to B
    - $X \leq S/2$
    - $X + Y = S$

$Y \geq X$



- **Proof c = ¾**
  - Case 2
    - Assume that k ≥ S/2 queries of A were assigned to B
    - At the time last query belonging to A was added to B, A's budget would have been greater or equal to B's budget
      - Otherwise the BALANCE algorithm **would not** make that assignment!
    - The only possibility is that some queries of B were assigned to A
    - Y is at least equal to k

Y ≥ X



Advertising: BALANCE Algorithm

- **Proof c = ¾**
  - Finally
    - X + Y = S
    - Y ≥ X
    - Worst case Y = X = S/2
    - Worst case revenue is S + S/2
    - Competitive ratio

$$c = \frac{\frac{3}{2}S}{2S} = \frac{3}{4}$$

Advertising: BALANCE Algorithm

- **For more then 2 advertisers the BALANCE algorithm has slightly lower competitive ratio – approx.. 0.63**

$$c = 1 - \frac{1}{e}$$

  - This appears to be the best solution for the Adwords problem – no online algorithm has a better competitive ratio!

Advertising: BALANCE Algorithm

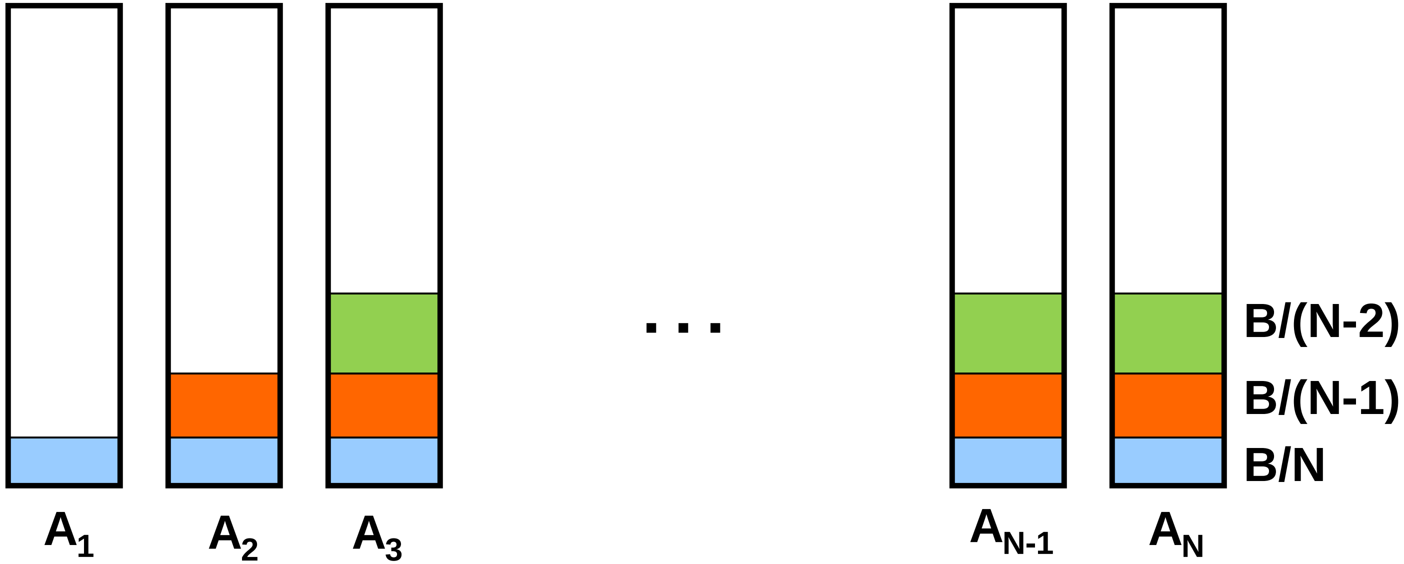
- **Worst case scenario $c = 1 - \frac{1}{e}$**
  - Let there be N advertisers $A_1, \ldots A_N$
  - Each advertiser has a budget B, B > N
  - N rounds, each containing B queries = N * B
  - Bidding
    - In i^{th} round, bids are placed by A_j where $j \geq i$
      - 1^{st} round $A_1, A_2, A_3, \ldots, A_N$ place bids
      - 2^{nd} round $A_2, A_3, \ldots, A_N$ place bids
      - …

Advertising: BALANCE Algorithm

- **Worst case scenario $c = 1 - \frac{1}{e}$**
  - Optimal ad matching
    - In the 1^{st} round all ads are assigned to A_1
      - It's the only round where A_1 has placed bids!
    - In the 2^{nd} round all ads are assigned to A_2
    - etc.
    - Total revenue is **N** · **B**

Advertising: BALANCE Algorithm

- Worst case scenario $c = 1 - \frac{1}{e}$



  - k[th] advertiser will have the following allocation after **k** rounds

$$S_k = \sum_{i=1}^{k} \frac{B}{N - i + 1}$$

Advertising: BALANCE Algorithm

- **Worst case scenario $c = 1 - \frac{1}{e}$**

$$S_k = \sum_{i=1}^{k} \frac{B}{N - i + 1}$$

  - At some point S_k will become greater than B
    - No further allocations are possible at that point
    - Advertisers with i < k have no active bids, and advertisers i ≥ k have to small budgets
  - Goal
    - Find smallest k so that S_k ≥ B

Advertising: BALANCE Algorithm

- **Worst case scenario** $c = 1 - \frac{1}{e}$

$$S_k = \sum_{i=1}^{k} \frac{B}{N - i + 1}$$

$\geq B$

$\boxed{B/N + B/(N-1) + \ldots + B/(N-(k-1))} + \ldots + B/3 + B/2 + B/1$

≥ 1

$\boxed{1/N + 1/(N-1) + \ldots + 1/(N-(k-1))} + \ldots + 1/3 + 1/2 + 1/1$

Advertising: BALANCE Algorithm

- **Worst case scenario $c = 1 - \frac{1}{e}$**

$$S_k = \sum_{i=1}^{k} \frac{B}{N - i + 1}$$

  - Proof by Euler:

~ 0 for larger k

$$\sum_{i=1}^{k} \frac{1}{k} = \ln(k) + \boxed{\gamma + \varepsilon_k}$$

1	ln(N - k)
1/N + 1/(N-1) + … + 1/(N-(k-1))	+ … + 1/3 + 1/2 + 1/1

Advertising: BALANCE Algorithm

- **Worst case scenario $c = 1 - \frac{1}{e}$**

$$S_k = \sum_{i=1}^{k} \frac{B}{N - i + 1}$$

$$\ln(N - k) = \ln(N) - 1$$

$$\Downarrow$$

$$\ln(N / (N - k)) = 1$$

$$\Downarrow$$

$$k = N(1 - 1/e)$$

- After k rounds no more ads can be assigned
  - Total revenue is $k \cdot B = B \cdot N(1 - 1/e)$

Advertising: BALANCE Algorithm

- **Worst case scenario $\boldsymbol{c = 1 - \frac{1}{e}}$**
  - Thus, the competitive ratio equals to:

$$c = \frac{NB\left(1 - \frac{1}{e}\right)}{NB} = 1 - \frac{1}{e}$$

Advertising: BALANCE Algorithm

- **Generally, advertisers do not have the same budget and do not place the same bids**
  - This fact ruins the performance of BALANCE algorithm

- **Example**
  - 2 advertisers A and B, 5 queries
  - A: budget 100, bid: 1
  - B: budget 80, bid: 10
  - BALANCE will select A and earn 5
  - Optimal revenue 50

Advertising: BALANCE Algorithm

- **Generalized BALANCE**
  - Having query **q** and bidder **I**
  - Bid value $\boldsymbol{x_i}$
  - Budget size $\boldsymbol{b_i}$
  - Amount spent so far = $\boldsymbol{m_i}$
  - Fraction of budget left over $\boldsymbol{f_i = 1 - \frac{m_i}{b_i}}$
  - $\psi_{\boldsymbol{i}}(\boldsymbol{q}) = \boldsymbol{x_i(1 - e^{-f_i})}$

Advertising: BALANCE Algorithm

- **Generalized BALANCE**
  - Taking into account CTR
    - CTR = c

$$\psi_i(q) = c \cdot x_i(1 - e^{-f_i})$$

  - Also worth considering historical frequency of queries

Advertising: BALANCE Algorithm

- **Generalized BALANCE**
  - Query *q* is assigned to the bidder *i* that has the largest value of $\psi_{\boldsymbol{i}}(\boldsymbol{q})$
  - Exhibits the same competitive ratio $\mathbf{1}-\frac{\mathbf{1}}{\boldsymbol{e}}$

Literature

1. J. Leskovec, A. Rajaraman, and J. D. Ullman, "Mining of Massive Datasets", 2014, Chapter 8 : “Advertising on the Web" (link)