Computational Advertising

Question 1: Suppose we apply the BALANCE algorithm with bids of 0 or 1 only, to a situation where advertiser A bids on query words x and y, while advertiser B bids on query words x and z. Both have a budget of $2. Identify a sequence of four queries that will certainly be handled optimally by the algorithm.

Sol:

yzyy is one sequence of four queries which yields the optimum i.e., $3.
yyxx is another sequence of four queries in which one advertiser is assigned to first x and the other is assigned to second x.

Question 2: The *set cover* problem is: given a list of sets, find a smallest collection of these sets such that every element in any of the sets is in at least one set of the collection. As we form a collection, we say an element is *covered* if it is in at least one set of the collection.

Note: In this problem, we shall represent sets by concatentating their elements, without brackets or commas. For example, {A,B} will be represented simply as AB.

There are many greedy algorithms that could be used to pick a collection of sets that is close to as small as possible. Here are some that you will consider in this problem.

Dumb: Select sets for the collection in the order in which they appear on the list. Stop when all elements are covered.

Simple: Consider sets in the order in which they appear on the list. When it is considered, select a set if it has at least one element that is not already covered. Stop when all elements are covered.

Largest-First: Consider sets in order of their size. If there are ties, break the tie in favor of the one that appears first on the list. When it is considered, select a set if it has at least one element that is not already covered. Stop when all elements are covered.

Most-Help: Consider sets in order of the number of elements they contain that are not already covered. If there are ties, break the tie in favor of the one that appears first on the list. Stop when all elements are covered.

Here is a list of sets:

<p align="center">AB, BC, CD, DE, EF, FG, GH, AH, ADG, ADF</p>

First, determine the optimum solution, that is, the fewest sets that can be selected for a collection that covers all eight elements A,B,...,H. Then, determine the sizes of the collections that will be constructed by each of the four algorithms mentioned above. Compute the ratio of the size returned by each algorithm to the optimum size.

Sol:

Given sets are: AB, BC, CD, DE, EF, FG, GH, AH, ADG, ADF

Dumb Method: AB, BC, CD, DE, EF, FG, GH → 7
Simple Method: AB, BC, CD, DE, EF, FG, GH → 7
Largest-First Method: ADG, ADF, AB, BC, DE, GH
→ 6 Most-Help Method: ADG, BC, EF, AH → 4

The optimum one is Most Help Method whose size is 4.
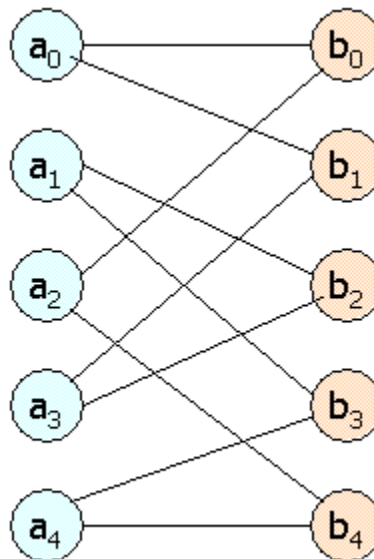
Ratio for Dumb and Most Help Methods = 7/4 = 1.75
Ratio for Simple and Most Help Methods = 7/4 = 1.75
Ratio for Largest First and Most Help Methods = 7/4 =
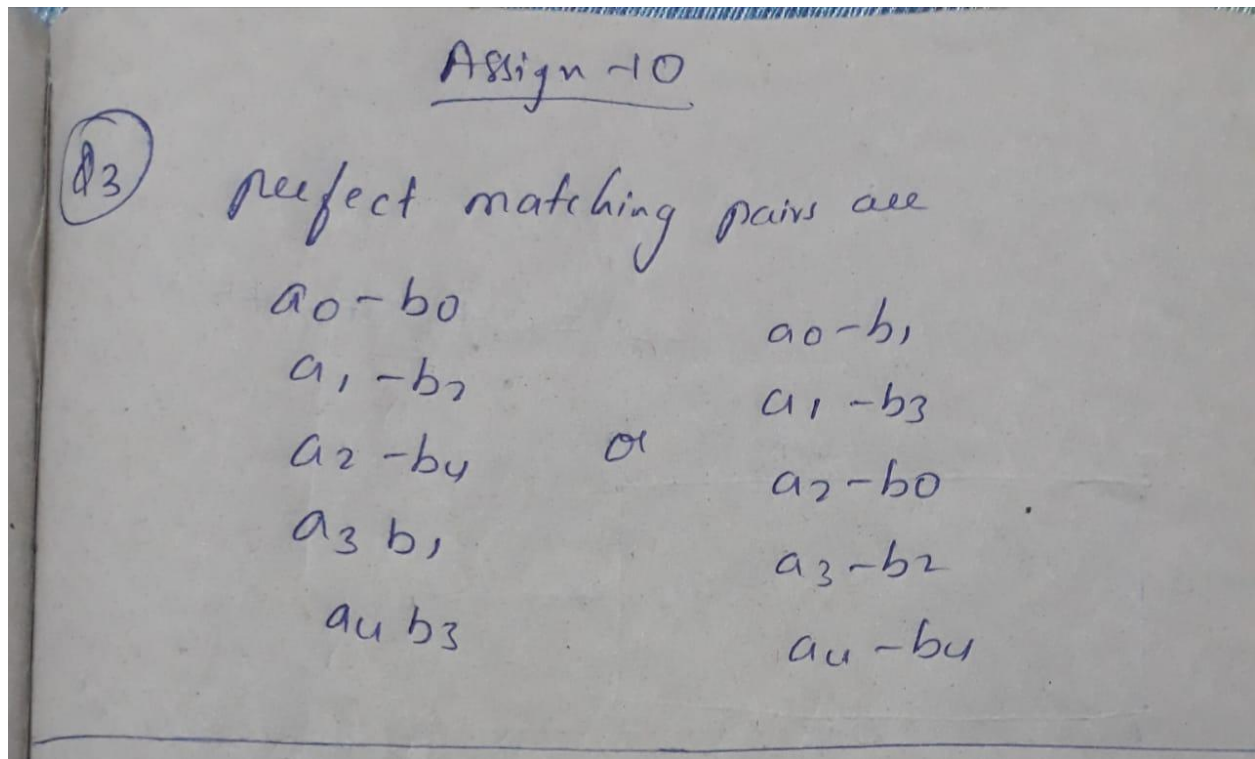1.75 Ratio for Most Help and Most Help Methods =
7/4 = 1.75

Question 3: This bipartite graph:

Has several perfect matchings. Find all the perfect matchings.

Sol:

Assign -10

(Q3) perfect matching pairs are

$a_0 - b_0$ $a_0 - b_1$

$a_1 - b_2$ $a_1 - b_3$

$a_2 - b_4$ or $a_2 - b_0$

$a_3 b_1$ $a_3 - b_2$

$a_4 b_3$ $a_4 - b_4$

Question 4: An ad publisher selects three ads to place on each page, in order from the top. Click-through rates (CTR's) at each position differ for each advertiser, and each advertiser has a different CTR for each position. Each advertiser bids for click-throughs, and each advertiser has a daily budget, which may not be exceeded. When a click-through occurs, the advertiser pays the amount they bid. In one day, there are 101 click-throughs to be auctioned.

Here is a table of the bids, CTR's for positions 1, 2, and 3, and budget for each advertiser.

Advertiser	Bid	CTR1	CTR2	CTR3	Budget
A	$.10	.015	.010	.005	$1
B	$.09	.016	.012	.006	$2
C	$.08	.017	.014	.007	$3
D	$.07	.018	.015	.008	$4
E	$.06	.019	.016	.010	$5

The publisher uses the following strategy to allocate the three ad slots:

1. Any advertiser whose budget is spent is ignored in what follows.
2. The first slot goes to the advertiser whose expected yield for the first slot (product of the bid and the CTR for the first slot) is the greatest. This advertiser is ignored in what follows.
3. The second slot goes to the advertiser whose expected yield for the second slot (product of the bid and the CTR for the second slot) is the greatest. This advertiser is ignored in what follows.
4. The third slot goes to the advertiser whose expected yield for the third slot (product of the bid and the CTR for the third slot) is the greatest.

The same three advertisers get the three ad positions until one of two things happens:

1. An advertiser runs out of budget, or
2. All 101 click-throughs have been obtained.

Either of these events ends one *phase* of the allocation. If a phase ends because an advertiser ran out of budget, then they are assumed to get all the clicks their budget buys. During the same phase, we calculate the number of click-throughs received by the other two advertisers by assuming that all three received click-throughs in proportion to their respective CTR's for their positions (round to the nearest integer). If click-throughs remain, the publisher reallocates all three slots and starts a new phase.

If the phase ends because all click-throughs have been allocated, assume that the three advertisers received click-throughs in proportion to their respective CTR's (again, rounding if necessary).

Your task is to simulate the allocation of slots and to determine how many click-throughs each of the five advertisers get.

Sol:

(Qu)

Advertiser	Bid×CTR 1	Bid×CTR 2	Bid×CTR 3
A	0.0015	0.001	0.0005
B	0.0014	0.00108	0.00054
C	0.0013	0.00112	0.00056
D	0.0012	0.00105	0.00056
E	0.0011	0.00096	0.0006
	slot 1	slot 2	slot 3

first slot ranking : A, B, C, D, E

Second " " : C, B, D, A, E

third " " : E, C, D, B, A

Since in slot 1, the expected revenue for A is higher i.e., 0.0015
In slot2, C is selected because 0.00112 is higher
In slot3, E is selected because 0.0006 is higher

The first phase ends when A gets 10 Click-throughs and now A runs out of budget and it is not eligible for second phase.
B is selected to first slot because 0.014 is higher than C, D, E.
Then C gets second slot and E gets third slot based on highest ones.

The second phase ends when B gets 22 Click-throughs and runs out of budget.

Now, in the third phase, C takes the first position and next second will be taken by D and third by E.

By ending of third phase, summing up Click-throughs for C, D, E in all phases, we get 36 click-throughs for C, 7 click-throughs for D, 26 click-throughs for E.

Click-throughs got for A are 10
Click-throughs got for B are 22
Click-throughs got for C are 36
Click-throughs got for D are 7
Click-throughs got for E are 26