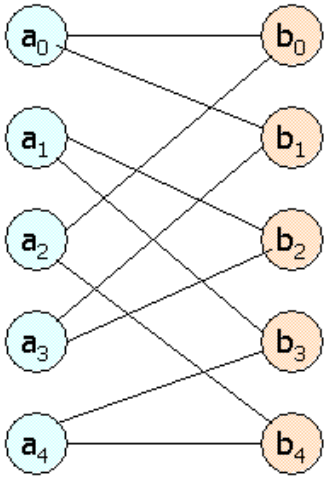
09: Computational Advertising

Number of questions:	4
Positive points per question:	3.0
Negative points per question:	1.0

Gradiance quiz on Computational Advertising. You can attempt to answer the questions as many times as you like. Questions get randomly regenerated each time. The score of the *last* submission gets saved into our records (that is, once you get a perfect score, don't submit again with a bad one).

1. This bipartite graph:



Has several perfect matchings. Find all the perfect matchings and then identify, in the list below, a pair of edges that can appear together in a perfect matching.

- a) a_0-b_1 and a_1-b_3
- b) a_2-b_0 and a_3-b_1
- c) a_2-b_4 and a_3-b_2
- d) a_0-b_0 and a_0-b_1

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

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:

AB, BC, CD, DE, EF, FG, GH, AH, ADG, ADF

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 the algorithm to the optimum size, and identify one of these ratios in the list below, correct to two decimal places.

- a) The ratio for **Most-Help** is 1.25
- b) The ratio for **Dumb** is 2.00
- c) The ratio for **Most-Help** is 1.33
- d) The ratio for **Most-Help** is 1.00

3. 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 in the list below a sequence of four queries that will certainly be handled optimally by the algorithm.

a) xyzx b) xzzz c) xyyx d) xxxy

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 who has budget in excess is ignored in what follows.

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.

- a) C gets 36 click-throughs.
- b) B gets 14 click-throughs.
- c) C gets 37 click-throughs.
- d) D gets 57 click-throughs.