princeton university fall '13 cos 521:Advanced Algorithms	
Homework 2	
Out: *Oct 13*	Due: *Oct 24*

You can collaborate with your classmates, but be sure to list your collaborators with your answer. If you get help from a published source (book, paper etc.), cite that. The answer must be written by you and you should not be looking at any other source while writing it. Also, limit your answers to one page, preferably less —you just need to give enough detail to convince the grader.

§1 Draw the full tree of possibilities for the cake-eating problem discussed in class, and compute the optimum cake-eating schedule. (If the tree is too large, you can draw it with the following slight change: The roommates eat 40% of the cake with probability 1/2, and the amount you eat each day has to be a multiple of 20%.)

§2 In class we designed a 3/4-approximation for MAX-2SAT using LP rounding. Extend it to a 3/4-approximation for MAX-SAT (i.e., where clauses can have 1 or more variables).

§3 (Game-playing equilibria) Recall the game of Rock, Paper, Scissors. Let's make it quantitative it by saying that the winning player gets \$ 1 from the other whereas a draw results in no exchange of money. Suppose we make two copies of the multiplicative weight update algorithm to play each other over many iterations. Both start using the uniformly random strategy (i.e., play each of Rock/paper/scissors with probability 1/3) and learn from experience using the MW rule. One imagines that repeated play causes them to converge to some kind of *equilibrium*. (a) Predict by just calculation/introspection what this equilibrium is. (Be honest; it's Ok to be wrong!). (b) Run this experiment on Matlab or any other programming environment and report what you discovered and briefly explain it. (We'll discuss the result in class.)

§4 (Supercommittee problem) There are n congressmen, and we are are given a list of m committees, each of which consists of a subset of the congressmen. We are trying to find the smallest *supercommittee*: a subset Z of congressmen such that there is at least one representative from each committee in Z. Design an $O(\log m)$-approximation to this problem. (Hint: Use LP rounding.)

§5 (Optimal life partners via MDP) Your friend is trying to find a life partner by going on dates with n people selected for her by an online dating servie. After each date she has two choices: select the latest person she dated and stop the process, or reject this person and continue to date. She has asked you to suggest the optimum *stopping rule*. You can assume that the n persons are all linearly orderable (i.e. given a choice between any two, she is not indifferent and prefers one over the other). The dating service presents the n chosen people in a random order, and her goal is to maximise the chance of ending up with the person that she will like the most among these n. (Thus ending up even with her second favorite person out of the n counts as failure; she's a perfectionist.) Represent her actions as an MDP, compute the optimum strategy for her and the expected probability of success by following this strategy.

(Hint: The Optimal rule is of the form: *Date γn people and decide beforehand to pass on them. After that select the first person who is preferable to all people seen so far.* You may also need that $\sum_{k=t_1}^{t_2} \frac{1}{k} \approx \ln \frac{t_2}{t_1}$.)

§6 In class we saw an algorithm using the basic MW rule that tries to manage a stock portfolio. Your goal is to implement it in any language (but Matlab or one of its free equivalents like Freemat may be easiest) and run it on 5 years of stock data to see how well it does. Include your code as well as the final performance (i.e., the percentage gain achieved by your strategy). The dataset is downloadable from the course homepage.

§7 (Firehouse location) Suppose we model a city as an m-point finite metric space with $d(x, y)$ denoting the distance between points x, y. The city has n houses located at points $v_1, v_2, \ldots, v_n$ in this metric space. The city wishes to build k firehouses and asks you to help find the best locations $c_1, c_2, \ldots, c_k$ for them, which can be located at any of the m points in the city. The *happiness* of a town resident with the final locations depends upon his distance from the closest firehouse. So you decide to minimize the cost function $\sum_{i=1}^{n} d(v_i, u_i)$ where $u_i \in \{c_1, c_2, \ldots, c_k\}$ is the firehouse closest to v_i. Describe an LP-based algorithm that runs in *poly*(m) time and solves this problem approximately. If OPT is the optimum cost of a solution with k firehouses, your solution is allowed to use $O(k \log n)$ firehouses and have cost at most $(1 + \epsilon)$OPT.

§8 (extra credit) In question 7 try to design an algorithm that uses k firehouses but has cost $O(\text{OPT})$. (Needs a complicated dependent rounding; you can also try other ideas.) Partial credit available for partial progress.