CS 573: Graduate Algorithms, Fall 2010
Homework 5

Due Friday, November 19, 2010 at 5pm
(in the homework drop boxes in the basement of Siebel)

1. Suppose we are given a set of boxes, each specified by their height, width, and depth in centimeters. All three side lengths of every box lie strictly between 10cm and 20cm. As you should expect, one box can be placed inside another if the smaller box can be rotated so that its height, width, and depth are respectively smaller than the height, width, and depth of the larger box. Boxes can be nested recursively. Call a box is *visible* if it is not inside another box.

    Describe and analyze an algorithm to nest the boxes so that the number of visible boxes is as small as possible.

2. Suppose we are given an array $A[1..m][1..n]$ of non-negative real numbers. We want to *round A* to an integer matrix, by replacing each entry x in A with either $\lfloor x \rfloor$ or $\lceil x \rceil$, without changing the sum of entries in any row or column of A. For example:

$$\begin{bmatrix} 1.2 & 3.4 & 2.4 \\ 3.9 & 4.0 & 2.1 \\ 7.9 & 1.6 & 0.5 \end{bmatrix} \longmapsto \begin{bmatrix} 1 & 4 & 2 \\ 4 & 4 & 2 \\ 8 & 1 & 1 \end{bmatrix}$$

    Describe an efficient algorithm that either rounds A in this fashion, or reports correctly that no such rounding is possible.

3. The Autocratic Party is gearing up their fund-raising campaign for the 2012 election. Party leaders have already chosen their slate of candidates for president and vice-president, as well as various governors, senators, representatives, city council members, school board members, and dog-catchers. For each candidate, the party leaders have determined how much money they must spend on that candidate's campaign to guarantee their election.

    The party is soliciting donations from each of its members. Each voter has declared the total amount of money they are willing to give each candidate between now and the election. (Each voter pledges different amounts to different candidates. For example, everyone is happy to donate to the presidential candidate,[1] but most voters in New York will not donate anything to the candidate for Trash Commissioner of Los Angeles.) Federal election law limits each person's total political contributions to $100 per day.

    Describe and analyze an algorithm to compute a donation schedule, describing how much money each voter should send to each candidate on each day, that guarantees that every candidate gets enough money to win their election. (Party members will of course follow their given schedule perfectly.[2]) The schedule must obey both Federal laws and individual voters' budget constraints. If no such schedule exists, your algorithm should report that fact.

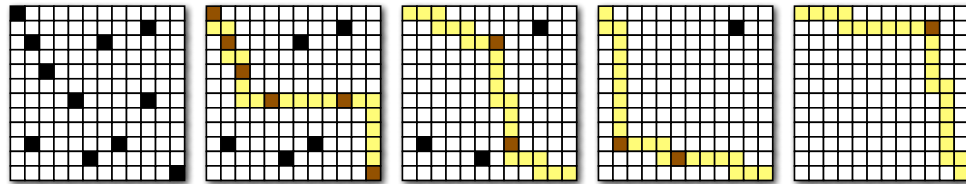[1] or some nice men in suits will be visiting their home.

[2] It's a nice house you've got here. Shame if anything happened to it.

4. Consider an $n \times n$ grid, some of whose cells are marked. A *monotone* path through the grid starts at the top-left cell, moves only right or down at each step, and ends at the bottom-right cell. We want to compute the minimum number of monotone paths that cover all the marked cells.

    (a) One of your friends suggests the following greedy strategy:

    - Find (somehow) one "good" path π that covers the maximum number of marked cells.
    - Unmark the cells covered by π.
    - If any cells are still marked, recursively cover them.

    Prove that this greedy strategy does *not* always compute an optimal solution.

    Greedily covering the marked cells in a grid with four monotone paths.

    (b) Describe and analyze an efficient algorithm to compute the smallest set of monotone paths that covers every marked cell. The input to your algorithm is an array $M[1..n, 1..n]$ of booleans, where $M[i, j] = \text{TRUE}$ if and only if cell (i, j) is marked.

5. Let G be a directed graph with two distinguished vertices s and t, and let r be a positive integer. Two players named Paul and Sally play the following game. Paul chooses a path P from s to t, and Sally chooses a subset S of **at most** r edges in G. The players reveal their chosen subgraphs simultaneously. If $P \cap S = \varnothing$, Paul wins; if $P \cap S \neq \varnothing$, then Sally wins. Both players want to maximize their chances of winning the game.

    (a) Prove that if Paul uses a deterministic strategy, and Sally knows his strategy, then Sally can guarantee that she wins.[3]

    (b) Let M be the number of edges in a minimum (s, t)-cut. Describe a deterministic strategy for Sally that guarantees that she wins when $r \geq M$, no matter what strategy Paul uses.

    (c) Prove that if Sally uses a deterministic strategy, and Paul knows her strategy then Paul can guarantee that he wins when $r < M$.

    (d) Describe a randomized strategy for Sally that guarantees that she wins with probability at least $\min\{r/M, 1\}$, no matter what strategy Paul uses.

    (e) Describe a randomized strategy for Paul that guarantees that he loses with probability at most $\min\{r/M, 1\}$, no matter what strategy Sally uses.

    Paul and Sally's strategies are, of course, algorithms. (For example, Paul's strategy is an algorithm that takes the graph G and the integer r as input and produces a path P as output.) You do *not* need to analyze the running times of these algorithms, but you must prove all claims about their winning probabilities. Most of these questions are easy.

---

[3]"Good old rock. Nothing beats rock. ... D'oh!"