Final Examination (take-home)

Handed Out: Monday, 30 April 2007
Due Date: Friday, 11 May 2007, 2:00pm.

ACADEMIC INTEGRITY STATEMENT

You may refer to the text, any handouts from this class, or notes you have taken during the class, but you should not use any other outside materials (e.g., other people, web searches). You may feel free to contact me for any clarification regarding the statement of the problems, however I do not intend to be very helpful toward the development of a solution. I am certainly willing to review any general topic that we have covered in the class.

INSTRUCTIONS

There are four problems overall, with the third problem being worth twice as much as the others. Given that this is a take-home exam, we expect your solutions to be legible and your explanations to be well written.

Problem #1 (20 points)

A local baker has come to you with the following problem. There are many orders that need to be filled, and for each item there is a *baking time*, b_i, and a subsequent *cooling time*, c_i. Each item must be baked and then cooled. Furthermore, only one item can be baked at any given time.

The baker must decide on the order for baking the items. That is the first item will be baked and then taken out of the oven to cool. The second item may be baked as soon as the first is taken out of the oven (even as that item is still cooling), and so on. The baker's goal is to minimize the overall time spent on the process (that is until all items have cooled). Your goal is to develop an efficient algorithm which produces a schedule whose completion time is as small as possible.

To help you organize your thought, we suggest the following three greedy rules:

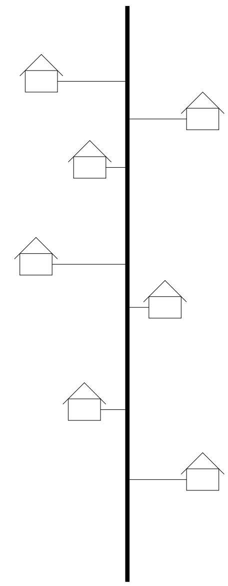
(a). Order the schedule based on b_i, from smallest baking time to largest baking time.

(b). Order the schedule based on c_i, from largest cooling time to smallest cooling time.

(c). Order the schedule based on the composite quantity $(b_i + c_i)$, ordered from smallest to largest.

One of these greedy rules is guaranteed to produce the optimal schedule; the other two rules are flawed. As a solution to this problem you should identify the proper rule and give a formal proof that it produces an optimal completition time. Furthermore, for each of the other two rules, you should produce a relatively simple instance to demonstrate the non-optimality of the result.

Problem #2 (20 points)



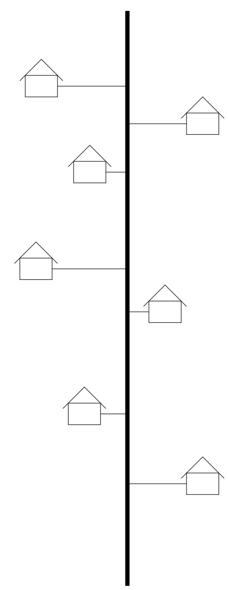
The local power company must connect all of its customers. Its plan is to build one main power line which runs **due North-South**. Then for each house, they will create a dedicated secondary line which runs due East or West to that house. The cost of the main power line will be fixed, as its length must go to the full North and South extremes of the customer base. However the cost of the secondary lines will vary depending upon the precise East/West alignment of the main line.

They have asked you to pick the optimal placement of the main North-South line, which minimizes this sum of secondary distances. (note: the figure to the right is for motivation; we are not necessarily suggesting that it is optimal)

Given a list of n houses, along with the (x, y) coordinates for each house, explain an efficient method to find the optimal placement of the main line.

You must prove that your approach guarantees the optimal placement.

Problem #3 (40 points)



The power company was very happy with our previous work in laying the North-South line. They want to go ahead with the plan as soon as possible. Having laid the main line, they now consider the need to lay the secondary lines which connect the n houses to the main line.
Assume the houses are already ordered from north to south as $h_1, h_2, \ldots, h_n$ and that the respective distances from the main line are $d_1, d_2, \ldots, d_n$. We presume that the time it takes a worker to lay a supplemental line is proportional to the distance. If one worker were to lay all of the lines, then the overall time required would be proportional to $D = \sum_{i=1}^{n} d_i$. However the power company has k workers which it can send out at once to divide up the work. In this way, the project will be complete as soon as the last of those workers is finished. The question is how to group the houses into k geographic regions so that a worker can be sent to each region to wire those houses.

As an example, if the houses are at distances $\langle 8, 12, 3, 7, 2, 10\rangle$ from the main line and $k = 3$, then the optimal division is $\langle 8\rangle$, $\langle 12, 3\rangle$ and $\langle 7, 2, 10\rangle$. This solution has cost 19 due to the time it takes to wire the third group. Notice that we are not allowed to rearrange the order of the houses, thus we cannot form groups such as $\langle 8, 7\rangle$, $\langle 12, 2\rangle$, $\langle 10, 3\rangle$. In this problem, we consider several approaches for finding the optimal solution. For simplicity, we will not be concerned about producing the actual grouping, but only in accurately computing the *cost* of the best possible solution.

(a). Rather than finding the optimal cost directly, consider a goal of determining whether or not it is possible to achieve a cost of C, for given constant C. Describe an efficient algorithm for this task. Justify the correctness and running time analysis.

(b). Describe an algorithm which determines the true optimal cost by making repeated use of the previous algorithm. Analyze the worst case running time.

(c). Develop an independent approach for solving the original problem based on the use of dynamic programming. In describing your solution, clearly explain:

- The set of subproblems which are used.
- A precise recursive formula used to solve a subproblem (including any necessary base cases)
- The order in which you will evaluate subproblems
- An analysis of the overall running time needed to solve the original problem.

Please try to make your algorithms as efficient as possible.

Problem #4 (20 points)

Let G be an arbitrary flow network, with a source s, a sink t, and a positive integer capacity c_e on every edge e. Let (A, B) be a minimum s-t cut with respect to these capacities. Consider the following two conjectures. For each you are to decide if it is true or false. If true, explain why; if false, give a simple counterexample.

(a). If we add 1 to every capacity, (A, B) remains a minimum s-t cut.

(b). If we double every capacity, (A, B) remains a minimum s-t cut.