Homework #2: Greedy Algorithms
Due Date: Wednesday, 8 March 2006

Problems

Problem A (10 points)

Consider the problem of making change for a certain quantity of currency (e.g., 82¢) using the fewest number of coins as possible. For this problem, we assume that any currency system includes a 1¢, so that it is always possible to make change for arbitrary amounts.

With the U.S. currency system (i.e., 100¢, 50¢, 25¢, 10¢, 5¢, 1¢ coins), it turns out that a simple greedy rule is provably optimal. Namely, starting with the biggest available denomination, take as many coins as possible so as not to bypass the stated goal. Then for the next biggest denomination, take as many coins as possible, and so on until reaching the exact target (e.g., for 82¢ we end up with 3 quarters, 1 nickel and 2 pennies, for a total of six coins).

However it turns out that this greedy rule is not guaranteed to be optimal for arbitrary currency systems. Give an example showing the failure of this rule. Namely:

- Define a coin system, $c_k > c_{k-1} > \ldots > c_1 > c_0 = 1¢$
- Give a specific target value (e.g., 82¢)
- List the solution which results from the greedy algorithm
- Give an alternate solution which uses strictly fewer coins than the greedy solution.

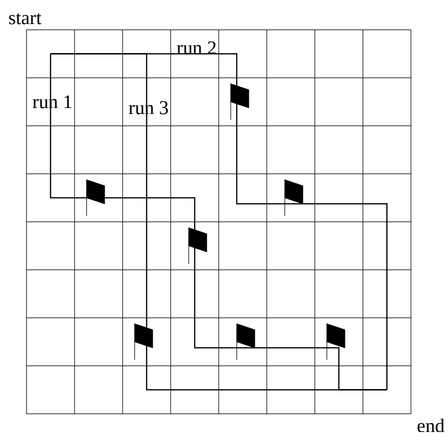
Problem B (10 points)

With the end of the Winter Olympics, we consider the following optimization problem. Someone is in charge of collecting all of the flags left interspersed on the ski slope. However they must do so by skiing by and grabbing them. Given that they cannot ski uphill, it may actually take more than one run to gather all the flags. We would like to be able to suggest a plan which uses as few runs as possible.

We model the problem as follows. The slope itself will be viewed as an $n \times n$ grid. The top of the ski-slope will be the top-left corner of the grid, and the hill is sloped so that a skier can move either downward or rightward at each individual step, until eventually ending up at the bottom-right corner of the grid. The flags will be placed at certain grid locations and we will be informed of the overall number of flags and the precise grid coordinates of the flags before we begin.

Someone has suggested the following greedy approach: When planning the first run, calculate a path which maximizes the number of flags which will be collected on that run (let's not worry about precisely how we calculate this local solution – just assume that we have a way to find such a path and use it). With those flags removed, repeat this approach, choosing a second run which maximizes the number of remaining flags which can be collected, and so on until collecting all flags.

An example of this approach on an instance is diagramed as follows.



Unfortunately, this algorithm does not always achieve a solution with the fewest number of overall runs. Demonstrate the inferiority of the plan as follows:

- Give an explicit instance of the problem, specifying the size of the grid and the exact placements of the flags within that grid.
- Clearly diagram the runs which you suggest would be chosen according the the greedy algorithm outlined.
- In a separate diagram, clearly identify a strictly smaller collection of runs which could have been used as a solution.

Problem C (10 points)
Exercise 2 of Chapter 4 of the text.

Problem D (10 points)
Exercise 5 of Chapter 4 of the text.

Problem E **(EXTRA CREDIT – 4 points)**

Describe an efficient algorithm which correctly solves the skiing cleanup problem. Clearly explain the algorithm as well as a proof that it produces an optimal solution.