CSCI 314 – Fall 2012 Handout: asgn04
Algorithms
Michael H. Goldwasser
Saint Louis University Wednesday, 26 September 2012

Homework #4: Greedy Algorithms
Due Date: Wednesday, 3 October 2012

Guidelines

Please make sure you adhere to the policies on collaboration and academic honesty as outlined in the syllabus.

Reading

Chapter 16 of CLRS.

Problems

There are three required problems worth a total of 100 points, and one extra credit challenge worth 10 points.

Problem A (30 points)

> **"Work entirely on your own."**
> CLR 16.2-4

Problem B (30 points)

> **"Work entirely on your own."**
> (See description on next page)

Problem C (40 points)

> **"You may discuss ideas with other students."**
> CLRS Problem 16-2

Problem D **(EXTRA CREDIT – 10 points)**

> **"You may discuss ideas with other students."**
> Describe an efficient algorithm that correctly solves the skiing cleanup problem introduced as Problem B. Clearly explain the algorithm as well as a proof that it produces an optimal solution.

Problem B (30 points)

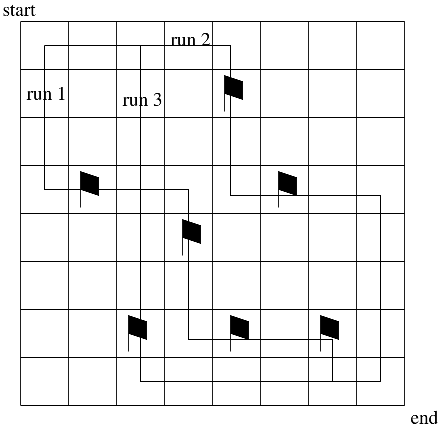
"Work entirely on your own."

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 that 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 that 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 that maximizes the number of *remaining* flags that 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 could 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.