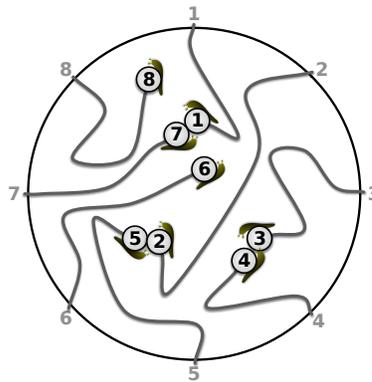


“CS 374” Fall 2014 — Homework 6

Due Tuesday, October 21, 2014 at noon

1. Every year, as part of its annual meeting, the Antarctic Snail Lovers of Upper Glacierville hold a Round Table Mating Race. Several high-quality breeding snails are placed at the edge of a round table. The snails are numbered in order around the table from 1 to n . During the race, each snail wanders around the table, leaving a trail of slime behind it. The snails have been specially trained never to fall off the edge of the table or to cross a slime trail, even their own. If two snails meet, they are declared a breeding pair, removed from the table, and whisked away to a romantic hole in the ground to make little baby snails. Note that some snails may never find a mate, even if the race goes on forever.



The end of a typical Antarctic SLUG race. Snails 6 and 8 never find mates.
The organizers must pay $M[3, 4] + M[2, 5] + M[1, 7]$.

For every pair of snails, the Antarctic SLUG race organizers have posted a monetary reward, to be paid to the owners if that pair of snails meets during the Mating Race. Specifically, there is a two-dimensional array $M[1..n, 1..n]$ posted on the wall behind the Round Table, where $M[i, j] = M[j, i]$ is the reward to be paid if snails i and j meet. Rewards may be positive, negative, or zero.

Describe and analyze an algorithm to compute the maximum total reward that the organizers could be forced to pay, given the array M as input.

2. Consider a weighted version of the class scheduling problem, where different classes offer different number of credit hours, which are of course totally unrelated to the duration of the class lectures. Given arrays $S[1..n]$ of start times, an array $F[1..n]$ of finishing times, and an array $H[1..n]$ of credit hours as input, your goal is to choose a set of non-overlapping classes with the largest possible number of credit hours.
 - (a) Prove that the greedy algorithm described in class — Choose the class that ends first and recurse — does *not* always return the best schedule.
 - (b) Describe an efficient algorithm to compute the best schedule.

In addition to submitting a solution on paper as usual, please *individually* submit an electronic solution for this problem on CrowdGrader. Please see the course web page for detailed instructions.

3. Suppose you have just purchased a new type of hybrid car that uses fuel extremely efficiently, but can only travel 100 miles on a single battery. The car’s fuel is stored in a single-use battery, which must be replaced after at most 100 miles. The actual fuel is virtually free, but the batteries are expensive and can only be installed by licensed battery-replacement technicians. Thus, even if you decide to replace your battery early, you must still pay full price for the new battery to be installed. Moreover, because these batteries are in high demand, no one can afford to own more than one battery at a time.

Suppose you are trying to get from San Francisco to New York City on the new Inter-Continental Super-Highway, which runs in a direct line between these two cities. There are several fueling stations along the way; each station charges a different price for installing a new battery. Before you start your trip, you carefully print the Wikipedia page listing the locations and prices of every fueling station on the ICSH. Given this information, how do you decide the best places to stop for fuel?

More formally, suppose you are given two arrays $D[1..n]$ and $C[1..n]$, where $D[i]$ is the distance from the start of the highway to the i th station, and $C[i]$ is the cost to replace your battery at the i th station. Assume that your trip starts and ends at fueling stations (so $D[1] = 0$ and $D[n]$ is the total length of your trip), and that your car starts with an empty battery (so you must install a new battery at station 1).

- Describe and analyze a greedy algorithm to find the minimum number of refueling stops needed to complete your trip. Don’t forget to prove that your algorithm is correct.
- But what you really want to minimize is the total *cost* of travel. Show that your greedy algorithm in part (a) does *not* produce an optimal solution when extended to this setting.
- Describe an efficient algorithm to compute the locations of the fuel stations you should stop at to minimize the total cost of travel.

CS 374 Fall 2014 — Homework 6 Problem 1

Name:	NetID:
Name:	NetID:
Name:	NetID:
Section:	1 2 3

Describe and analyze an algorithm to compute the maximum total reward that the SLUG Mating Race organizers could be forced to pay, given the reward array M as input.

CS 374 Fall 2014 — Homework 6 Problem 2

Name:	NetID:
Name:	NetID:
Name:	NetID:
Section:	1 2 3

-
- (a) Prove that the greedy algorithm described in class — Choose the class that ends first and recurse — does *not* always return the valid schedule with the maximum number of credit hours.
- (b) Describe an efficient algorithm to compute the best schedule.
-

CS 374 Fall 2014 — Homework 6 Problem 3

Name:	NetID:
Name:	NetID:
Name:	NetID:
Section: 1 2 3	

-
- (a) Describe and analyze a greedy algorithm to find the minimum number of refueling stops needed to complete your trip. Don't forget to prove that your algorithm is correct.
 - (b) But what you really want to minimize is the total *cost* of travel. Show that your greedy algorithm in part (a) does *not* produce an optimal solution when extended to this setting.
 - (c) Describe an efficient algorithm to compute the locations of the fuel stations you should stop at to minimize the total cost of travel.
-