CSC 373H (2007): Assignment 4

Worth 5%. Due August 9 at 6pm in lecture.

The work you submit must be your own. You may discuss problems with each others; however, you should prepare written solutions alone. Copying assignments is a serious academic offence, and will be dealt with accordingly.

Question 1 [Greedy Algorithm]

Given a nonempty set S of n elements. A nonempty family $\mathcal{L}$ of subsets of S is called *nice* if it satisfies the following conditions:

1. **Inclusion property**: For every subsets $A, B \in S$, if $A \subseteq B$ and $B \in \mathcal{L}$ then $A \in \mathcal{L}$. (In other words, if B is a member of $\mathcal{L}$ then all subsets of B are also members of $\mathcal{L}$. Note that the empty set $\varnothing$ is necessarily a member of $\mathcal{L}$.)
2. **Exchange condition**: If $A \in \mathcal{L}$ and $B \in \mathcal{L}$ and $|A| < |B|$ (here $|A|$, $|B|$ denotes the number of elements in A and B respectively), then there is some element $x \in B - A$ such that $A \cup \{x\} \in \mathcal{L}$.

A subset A in $\mathcal{L}$ is called a *top* set if there is no other set B in $\mathcal{L}$ such that $A \subset B$.

Given a set $S = \{1, 2, \ldots, n\}$ where each element i has a weight $w(i)$, and a nice family $\mathcal{L}$ of subsets of S. The weight of a set A in $\mathcal{L}$ is the total weight of the elements in A:

$$w(A) = \sum_{i \in A} w(i)$$

The problem is to find a set A in $\mathcal{L}$ with maximum weight. Notice that any set A in $\mathcal{L}$ of maximum weight must be a top set. Also, $\mathcal{L}$ might have as many as 2^n members. Therefore going through every member of $\mathcal{L}$ is not an option here, because the problem can be solved in polynomial time using the greedy approach.

a) Give a Greedy algorithm that finds a maximum weight subset in $\mathcal{L}$. Prove that your algorithm is correct.

b) To analyze the running time of your algorithm, we assume that checking whether a subset $A \subset S$ is a member of $\mathcal{L}$ takes time $t(n)$. (This checking time only depends on n—the number of elements in S—and not on the set A.) What is the running time of your algorithm in terms of n and $t(n)$ (state your answer using $\mathcal{O}$ notation)?

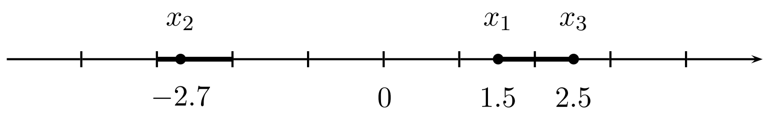
Question 2 [Greedy Algorithm]

Consider the problem that, given a set $\{x_1, \ldots, x_n\}$ of points on the real line, determines the smallest set of unit-length closed intervals that contains all of the given points.

For example, on input $\{-2.7, 1.5, 2.5\}$, the following set of two unit intervals is an optimal solution:

$$\{[-3, -2], [1.5, 2.5]\}$$

a) Give the pseudo-code for a greedy algorithm that solves the above problem and prove that your algorithm is correct.



b) What is the running time of your algorithm?

Question 3 [Approximation Algorithme]