MCS-375: Algorithms: Analysis and Design Handout #GS
San Skulrattanakulchai
Gustavus Adolphus College Oct 30, 2014

Study Questions for Greedy Algorithms

1. For the activity scheduling problem, give a set of classes, with their start and finish times, such that there are 2 optimum schedules—the first schedule can be constructed by the greedy algorithm, but the second schedule cannot.

2. Give a set of 4 classes such that the schedule obtained from earliest-finish-time-first algorithm differs from that obtained from latest-start-time-first algorithm.

3. Let S be an earliest-finish-time-first schedule for a collection of classes $\mathcal{C}$. Each class in $\mathcal{C}$ starts and ends at some hour of the day, e.g. 3:00, not 3:20.

   (i) The administration says that now every class of C must end exactly 10 minutes earlier. No start times change. Is S still a valid schedule, and if so is it optimal?

   (ii) The administration changes its mind. Each class must add an additional 10 minutes to its original ending time (for cleanup). No start times change. Is S still a valid schedule, and if so is it optimal?

4. Give one set of classes such that none of these greedy algorithms give an optimal schedule:

   - shortest-time first
   - longest-time first

5. Give pseudocode to implement the shortest-time-first algorithm in time $O(n^2)$. (Can you give one in time $O(n \log n)$?)

6. A set of items has weights 1, 2, 4, 5, 6.

   (i) Draw a tree with minimum w.e.p.l. constructed by Huffman's algorithm.

   (ii) Draw a tree for these items having minimum w.e.p.l. but cannot be constructed by Huffman's algorithm.

7. Professor Dull claims that If B is a binary tree such that swapping any of its two leaves increases the w.e.p.l. or keeps it the same, then B has minimum w.e.p.l. Give a counterexample to Dull's claim.

8. Given a set of items I with nonnegative weights, we want to find a binary tree having I as leaves and having **maximum** w.e.p.l.

(i) Give a greedy algorithm for the problem.

(ii) State a theorem that would prove this algorithm is correct.

(iii) Give a swapping argument that proves the theorem.