COMP 360 Algorithm Design Techniques

Assignment 2: Greedy algorithms; BFS and DFS

Tutorials will be held Tues. and Wed., Sept. 30 and Oct. 1. Reading assignments are on the course web site, on the approximate schedule page.

Due date: Friday, Oct. 3, noon.

1. *Counter-examples for Greediness.*

   (a) Consider a ruler with $n = 2^m$ links. Suppose we fold this ruler by recursively finding the best way to fold links $1, 2, \ldots, n/2$ (with the other links removed, and the best way to fold links $n/2+1, n/2+2, \ldots, n$; then we place the two optimally folded subrulers together in the way that gives the best total folded length. Find an example to show that this algorithm will not always give the optimum folding.

   (b) For the activity scheduling problem, let $c(i)$ denote the number of other activities that activity i conflicts with. Suppose we order the activities in order of increasing $c(i)$, and process the ordered list as follows. We take each item on the list in turn, and add it to the schedule if and only if it does not conflict with any activity that is already on the schedule. (We do not recompute the $c(i)$ at each step.) Find an example to show that this does not always give an optimum solution.

2. *Weighted Activity Selection.*

   Consider a modified version of the activity selection problem in which each activity is given a non-negative weight (or value). Suppose our goal is to find a feasible schedule of maximum total weight. Give examples where the following greedy algorithms fail to produce the optimal solution.

   (a) At each step select a feasible activity with earliest finish time.

   (b) At each step select a feasible activity of maximum value.

3. *Two-Colourability*

   A graph G *2-colourable* (or *bipartite*) if every vertex of G can be coloured either red or blue in such a manner that every edge has one end-vertex coloured red and the other coloured blue.

   (a) Prove that any graph that contains a cycle of odd length (i.e. a cycle with an odd number of vertices) is not 2-colourable.

   (b) Prove that any graph that does not contain a cycle of odd length is 2-colourable. [*Hint.* Grow a breadth first search tree in G and use it to colour the vertices red and blue.]

4. *Depth First Search*

   [Notation: let $start(x)$ denote the time when vertex x is first examined in a depth first search tree i.e. the time x is added to the stack.]

   Suppose there is a path from u to v in a directed graph G. Give a counter-example to the following conjecture: if $start(u) < start(v)$ in a depth first search tree T of G, then u is an ancestor of v in T.

5. *The Dépanneur's Problem*

   Work out the details of the following problem, begun in class. There is a knapsack of capacity W and n items $\{1, 2, \ldots, n\}$. Item i, $1 \leq i \leq n$, has value v_i and weight w_i. (Assume that W, w_i and v_i are non-negative integers). In addition, suppose that there are an unlimited number of copies of each item. Subject to the constraint that the total weight of the selected items is at most W, how many copies (the number of copies must be a non-negative integer) of each item should we place in the knapsack in order to maximize the total value of the selected items?

   Propose a dynamic programming solution.

   Illustrate your algorithm at work on a real, non-trivial example. Make up your own data, and compute the table.