

University of Delaware -- Computer and Information Science

CIS621 -- Homework 2

Handed out: September 15, 2011
Due date: September 29, 2011

4. [*Individual Problem*]

[Problem 4.11 from Aho, Hopcroft and Ullman]

Consider the following two operations defined on a subset S of the integers from 1 to n :

- **Select:** returns any member of S and deletes that member from S
- **Add(i):** makes integer i a member of S . If i is already a member of S , then nothing is done.

Devise an efficient data structure to represent S such that **Select** and **Add** operations run in time $O(1)$. Initialization of the data structure is also restricted to $O(1)$ time. You may assume that the allocation of space for an array of a specified size occurs in time $O(1)$.

5. [*Group Problem*]

Give an efficient implementation of an operation **FIB-HEAP-PRUNE**(H, r), which deletes an arbitrary $\min(r, n[H])$ nodes from Fibonacci Heap H . Here, "arbitrary" should be taken to mean "your algorithm may delete any $\min(r, n[H])$ nodes that it desires".

Analyze the running time of your implementation. Once you have a solution that you are convinced is absolutely the fastest possible amortized running time, then you will need to check with Professor Lloyd to see if he agrees.

6. [*Group Problem*]

Consider a rooted tree with nodes labeled 1, 2, 3, ..., n . Devise a method for determining whether, given two arbitrary nodes in the tree, one node is an ancestor of the other.

Your method should use time $O(1)$ per input pair in the worst case.

Note: Your method will rely on some information that you have gathered in advance about the tree -- the time to gather this information must be $O(n)$. That's alright -- that setup time is not counted when determining the running time for an input pair. The idea is that you do the setup just once, and then answer "queries" many, many times, so that the relevant cost is not the cost of the operations that you do once (the setup), but rather the cost of the operation done over and over (the query).