INFO6205-03 Final Exam Spring 2018

Introduction

This document just covers the four questions that must be manually graded. If you have queries about the other questions, please let me know (Slack #general).

Question 4

In this question, which has four parts, you have to consider simple search strategies where you have a predetermined list of *M* elements, and expect to perform *N* searches.

- The simplest search mechanism is to create a linked list, inserting new elements at the head of the list and scanning the list (using *equals*) when searching. What is the average search cost (including the cost of setting up the list ahead of time)? Assume, for the sake of the question, that the cost of testing equality is the same as the cost of inserting an element in the list. [unit cost]
- If N is much larger than M, what is the average cost per search?
- If the key of an element has an ordering available, show a simple improvement is to sort the list and then use binary search. What is the average cost per search now?
- If N is much larger than M, what is the average cost per search now?

Answer:

- M/N + M/2
- M/2
- (M/N + 1) lg M
- lg M

Observations

This was supposed to be a really easy question, but surprisingly, many of you had trouble with it. It's possible that you didn't understand the question properly. Many turned it immediately into something about O(N), but that wasn't what was asked. There were only two points available for each part of the answer. I tried to be generous/understanding where you misinterpreted the question. Some of you also mixed up the parts, especially parts 3 and 4. Again, I did my best to credit you for the right ideas.

Question 5

- Give an example of a *reduction*.
- Note the costs incurred by the reduction.
- How do these costs compare with alternative algorithms?

Sample answer:

a) The problem of finding the largest element in a list reduces to sorting the list and taking the first element
b) 0 + n lg n + constant = n lg n
c) n (scanning each element)

Observations

Again, there was some confusion about this question. In particular, many students didn't realize that parts b and c were based on part a! I was not asking you to copy text from books or slides. I needed you to interpret those questions regarding a specific example (the one you chose in part a). I did my best to credit you with the right ideas.

Question 9

1. In a depth-first-search-based algorithm, what is meant by "reverse post-order?"
2. Name an algorithm which uses reverse-post-order.
3. What standard data structure is required for reverse-post-order?

Answer:

1. Post-order is the order of vertices resulting from noting the value of a vertex *after* the the algorithm has recursed through the adjacent vertices. Reverse post-order is the opposite of post-order.
2. Topological sort (or Kruskal's algorithm).
3. Stack.

Observations

Again, some students misinterpreted this question, especially the “what is meant by” part. However, most people did get the right answer for part 3.

Question 10

Explain, using key points, how a binary heap works, how it differs from a binary tree and what makes it so suitable as the underlying data structure of a priority queue?

Answer:
A binary heap is essentially an array [2] and is indexed rather than pointed to (as in a tree) [1.5]; the children of node *k* are always found at *2k* and *2k+1* [1]; the key of node *k* is always greater than (or less than) the keys of each child [1.5] whereas in a binary tree the key of the parent would (typically) be intermediate [0.5]; thus the maximum (or minimum) key is always found *at the root* [2]. A binary heap is a “complete” binary tree [1.5]. Average time to insert is O(1) [1] but worst-case time is O(lg n) [1.5]. It’s suitable because of its efficient and balanced insert and deleteMax methods—worst case O(lg n), cache behavior [1]. Bonus for any reasonable answer: [1.5]. Total points possible: 15 including 2.5 bonus points.

Observations

This is going to be a hard question to grade. In retrospect, there was too much that could be said here. So far, most students are getting pretty good marks for this question, though none are perfect so far.