Selected Solutions for Written Midterm

CSCI2100B

April 23, 2013

Problem 1(b)

Solution:

 $f_1(n) = 100 \log n, f_2(n) = 3n.$

When n = 256, $f_1(n) = 800$ and $f_2(n) = 768$, which leads to $f_1(256) > f_2(256)$; when n = 512, $f_1(n) = 900$ and $f_2(n) = 1536$, which leads to $f_1(512) < f_2(512)$. Thus, n is between 256 and 512. 300 is a pretty good estimate for n.

Problem 2(b)

Solution:

This problem is the most difficult one in the exam. I hope this solution can make you understood step by step.

For stacks, we first use F(n) to denote the number of valid final sequences. We separate the situations according to the position of 1. If 1 appears in the first place in the final sequence, it means that 1 does not go into the stack (or push then immediately pop). In this case, there are totally F(n-1) possible sequences since $2, 3, \ldots, n-1$ will go into the stack afterwards. If 1 appears in the *i*-th place, which indicates that $2, 3, \ldots, n-1$ will go into the stack afterwards. If 1 appears in the *i*-th place, which indicates that $2, 3, \ldots, n-1$ will go into the stack afterwards. If 1 appears in the *i*-th place, which indicates that $2, 3, \ldots, i$ are in the first i-1 positions (Why? You can imagine what will happen if a number larger than *i* exists in the first i-1 position) and $i+1, \ldots, n$ are in the last n-i positions. In this case, there are totally F(i-1)*F(n-i) possible sequences. By iterating *i* from 1 to *n*, we can obtain the recursive formula $F(n) = \sum_{i=1}^{n} F(i-1) \cdot F(n-i)$. It is easy to obtain that F(0) = 1, F(1) = 1, F(2) = 2, F(3) = 5. F(7) can be easily calculated after a few steps. Actually, this series of numbers is called **Catalan Numbers**. The formula for *n*-th Catalan number is $\frac{1}{n+1} {n \choose n}$.

For queues, the method above is not applicable because we cannot separate the situations according to the position of either 1 or n. Actually, the basic idea is to build an one-to-one mapping from the final sequences of queue to the final sequences of stack. For every valid operation sequence (e.g. IISSOISOO for n = 6), we can apply it to both stack (in this example is 342651) and queue (in this example is 341625) to get two different final sequences. Thus, we maps the final sequence of stack to the final sequence of queue. By constructing this mapping, we can conclude that the answer for queue is the same as stack for any n. Unfortunately, there are still some unsolved issues. The first is why the sets of operation sequences for stack and queue have the same elements. The second is why for every final sequence the operation sequence (valid means NumberOf(S) + NumberOf(I) = n, NumberOf(I) = NumberOf(O) and for every subsequence starting from beginning $NumberOf(I) \leq NumberOf(O)$). Thus, they are the same. For the second issue, one of your programming assignment can prove it since you can design an algorithm to generate the unique one.

Problem 3(c)(d)

Solution:

The answer for 3(c) is $2^{n+1} - 1$ because the root will be the last one in post-order traversal. The answer for 3(d) is 2^n because the root will be precisely in the middle when performing in-order traversal.

Problem 4(e)

Solution:

If you do problem (a) right, the answer should be 15 and 20, OR 15 and 22, OR 20 and 22. Then you need to perform a double rotation to put 27 as the root.

Problem 5(j)

Solution:

We've discussed about the method to deal with the sequence in the form of T(n) = aT(n-1)+b, which is to let S(n) = T(n) + k and make S(n) = aS(n-1). Thus we can solve k using a and b.

Here the problem becomes more difficult. Actually only one more item is needed to use this method. Let S(n) = T(n) + kn + t and make S(n) = aS(n-1). Then T(n) = aT(n-1) + bn becomes S(n) - kn - t = a(S(n-1) - k(n-1) - t) + bn. After adjusting, it becomes S(n) = aS(n-1) + (b - ak + k)n + ak - at + t. To make it S(n) = aS(n-1), we obtain two equations. One is b - ak + k = 0 and the other is ak - at + t = 0. By solving them, we get $k = \frac{b}{a-1}$ and $t = \frac{ab}{(a-1)^2}$.

However, to we still need to consider the situation that a = 1, which is easy to handle.