CS 303 Algorithms — Prof. Lewis
due mon sep 24, 2014

H4.

Yingzhu (Jacqueline) Zhang

Students I have consulte with for this assignment:
1. William Bench
2. Nathaniel Marshall
3. Jacob Shor
4. William Theuer
5. Maurice Hayward
6. Stephanie Bramlett
7. Diego Lanao
8. Meade Nelson
9. Kalyn Horn

1. 2.3.4. **Solution:**

   $Example1$: A B C D E F G H I J
   $Example2$: K L M N O P Q R S T
   $Example3$: 1 2 3 4 5 6 7 8 9 10
   $Example4$: 11 13 15 17 19 21 23 25 27 29
   $Example5$: 10 9 8 7 6 5 4 3 2 1
   $Example6$: J I H G F E D C B A

2. 2.3.10. **Solution:**

   Given: $N = 10^6, .1N^2 = 10^11$
   Also, according to Proof of Proposition L, the standard deviation for quicksort is $.65N$
   And we know that the mean of quicksort compares is $2NlnN$
   $\therefore k = \frac{.1N^2-2NlnN}{.65N}$
   $\therefore$ Probability $P = \frac{1}{\frac{10^11-2(10^6)ln(10^6)}{.65(10^6)}} \approx 4.2273 \times 10^{-11}$

3. 2.3.13. **Solution:**

   Because the recursive call the quicksort performs is when it performs $partition()$; Also because a stack contains pertinent information for each recursive call. When a procedure is invoked, its information is pushed onto the stack; when it terminates, its information is popped
   $\therefore$
   best case: lgN ($partition()$ splits the array exactly in half)
   worst case: $N-1$ ($partition()$ splits the array as unevenlly as possible)
   best case: lgN ($partition()$ falls in the middle on average)

4. Give the heap that results when the keys P O S S U M L O A F are inserted in that order into an initially empty max-oriented heap.
   Solution:

   See attached sheets of paper for the drawing.

5. 2.4.7. **Solution:**

   when $k = 2$, kth largest position can appear in the the leaves of height 2; cannot appear in any other leaves
   when $k = 3$, kth largest position can appear in the leaves of heights 2 and 3; cannot appear in any other leaves
   when $k = 4$, kth largest position can appear in the leaves of heights 2, 3 and 4; cannot appear in any other leaves
   See attached sheets of paper for an illustration of the height of the heap.

6. Give an example that shows that quicksort is not stable.
   Solution:

   Suppose we have an array containing $F\ O^1\ O^2\ D\ O^3\ O^4$
   Using quicksort, our sorted result would be:
   $FD\ O^2\ O^1 D\ O^3\ O^4$
   $DF\ O^4\ O^1 D\ O^3\ O^2$
   However, if we were to use a stable sort method, such as mergesort, the result would be:
   $DF\ O^1\ O^2 D\ O^3\ O^4$
   $\therefore$ We can see here that quicksort is not stable because it does not account for the different values associated with the same entry.

7. 3.3.1. **Solution:**

   See attached sheets of paper for the drawing.

8. 3.3.3. **Solution:**

   In order to find the insertion order, we can first sort the key in ascending order, and we get
   $ACEHMRSX$
   Then, we draw a 2-3 tree whose height is one (see attached paper);
   Then, we try to draw out the 2-3 tree which results this array of keys step-by-step (see attached paper) and collect the insertion order;
   Then we get the insertion order as:
   $RAXHSECM$

9. 3.3.4. **Solution:**

   Given a list of N objects, when we contruct for a tree that is all 2-nodes, this tree of height h must have at most $2^d + 1$ leaves
   Similarly, when we constrct for a tree that is all 3-nodes, this tree of height h must have at most $3^d + 1$ leaves
   $\therefore$ We have an equation:
   $N \leq 2^d$ or $N \leq 3^d$, which is $lgN \leq h$ or $log_3 N \leq h$
   $\therefore$ It is true that the height of a 2-3 tree with N keys is between $\sim \lfloor\ log_3 N\ \rfloor$ (for a tree that is all 3-nodes) and $\sim \lfloor\ lgN\ \rfloor$ (for a tree that is all 2-nodes).