22000 Algorithms, Summer 2004, CCNY CUNY

Instructor: Natalia Novak

Midterm Exam, June 30, 2004

1. Draw a recursion tree to determine a good assymptotic upper bound on the recurrence T(n) = 3T($\frac{n}{3}$) + n. Use the substitution method to verify your answer.

   solution: see mt1.jpg (the link can be found on the course's **Notices** page).

2. Find an assymptotic upper bound on the summation

$$\sum_{k=0}^{n} \frac{(k+1)^2}{4^k}$$

   by splitting summations method.

   solution: let's find a ratio r such that $\frac{\frac{(k+2)^2}{4^{k+1}}}{\frac{(k+1)^2}{4^k}} \leq r$, (where $0 < r < 1$).
   $\frac{\frac{(k+2)^2}{4^{k+1}}}{\frac{(k+1)^2}{4^k}} = \frac{(k+2)^2}{4^{k+1}} \cdot \frac{4^k}{(k+1)^2} = \frac{(k+2)^2}{4(k+1)^2}$
   if k = 0 then $\frac{(k+2)^2}{4(k+1)^2} = \frac{4}{4} = 1$
   if k = 1 then $\frac{(k+2)^2}{4(k+1)^2} = \frac{9}{16}$
   So $\frac{\frac{(k+2)^2}{4^{k+1}}}{\frac{(k+1)^2}{4^k}} \leq \frac{9}{16}$, for k> 0. Let's take r = $\frac{9}{16}$
   Hence

$$\sum_{k=0}^{n} \frac{(k+1)^2}{4^k} \leq \sum_{k=0}^{\infty} \frac{(k+1)^2}{4^k} = 1 + \sum_{k=1}^{\infty} \frac{(k+1)^2}{4^k} \leq 1 + \frac{4}{4} \cdot \frac{1}{1 - \frac{9}{16}} = 1 + \frac{16}{7} = 3\frac{2}{7}$$

   Thus

$$\sum_{k=0}^{n} \frac{(k+1)^2}{4^k} = O(1)$$

3. Is the sequence $A =< 20, 10, 15, 3, 6, 9, 8, 1, 2, 4, 8, 5 >$ a heap (max-heap in terms of new book)? (explain your answer)

   solution: the given sequence is not a heap. The explanation is presented in mt3.jpg (link to mt3.jpg can be found on course's **Notice** page).

4. What is the effect of calling Heapify(A,i) (Max-Heapify(A,i) in terms of new book) when the element A[i] is larger than its children?

   solution: no effect on array A. Since when the Heapify(A,i) is called, it is assumed that the binary trees rooted at Left(i) and Right(i) are heaps, but A[i] may be smaller than its children, thus violating the heap (or max-heap) property, so Heapify(A,i) fixes it. In our case element A[i] is larger than its children, thus heap (or max-heap) property is not violated and call to Heapify(A,i) won't do anything with the array A.

5. Illustrate the operation of Quicksort on the array $A = \{13, 5, 10, 22, 11, 4, 8, 16, 1, 19, 30, 3\}$.

   solution: see mt7.jpg (the link to mt7.jpg can be found on course's **Notice** page)

6. Write a pseudocode of an algorithm that, given n integers in the range 0 to k, preprocesses its input in O(n+k) time and then answers any query about how many integers fall into a range $[a \dots b]$ (i.e. the number of elements A[i] such that a$\leq$ A[i] $\leq$ b) in O(1) time (assume that $0 \leq a \leq b \leq k$).
   Hint: recall the Counting sort

   solution:

```
integers-in-the-range(A,k,a,b)
1. for i <- 1 to k
2.     do C[i] <- 0
3. for j <- 1 to length[A]
4.     do C[A[j]] <- C[A[j]]+1
5. > C[i] now contains the number of elements equal to i.
6. for i <- 2 to k
7.     do C[i] <- C[i] + C[i-1]
8. > C[i] now contains the number of elements less than or equal to i
9. if a >= 0 and b <= k
10.   then if a = 0
11.           then print C[b]
12.           else print C[b] - C[a-1]
13.   else error"a,b are out of the range"
```

   lines 1-8 are from Counting-sort pseudocode - here the given integers (from the input array A) are preprocessed. It takes O(n+k) time (since we know that the running time of Counting-sort is O(n+k))
   lines 9-13 for given a and b the number of elements which fall into the range [a...b] is printed. It takes O(1) time.

7. Suppose that we have numbers between 1 and 1000 in a binary search tree and want to search for the number 363. Which of the following sequences **could not** be the sequence of nodes examined? Explain your answers.
   a. 2,252,401,398,330,344,397,363
   b. 924, 220, 911, 244, 898, 258, 362, 363
   c. 925, 202, 911, 240, 912, 245, 363
   d. 2, 399, 387, 219, 266, 382, 381, 278, 363
   e. 935, 278, 347, 621, 299, 392, 358, 363

solution: **a.** can be the sequence of nodes examined
b. can be
c. could not be
d. can be
e. could not be.
See mt6.jpg for explanations (the link to mt6.jpg can be found on course's **Notice** page).