Mathematical Analysis of Algorithms

**Homework #10** (Last Set)
**Due Date:**
**Reading Assignment:** 6.5, 9.5–9.6
**Problems:**

1. 9–7

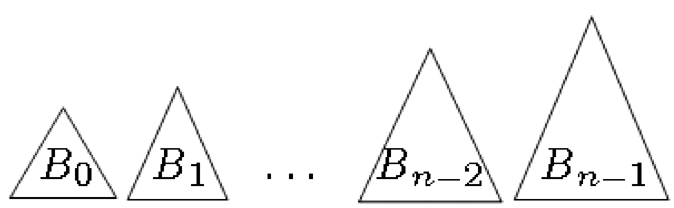
2. 9–32

3. 9–35

4. 9–36

5. A binomial tree is a tree having a special structure (irrelevant to this problem) which guarantees that the tree has $2^k$ nodes for some $k \geq 0$; a binomial tree with $2^k$ nodes is called a $B_k$ tree. Each node in a $B_k$ tree contains a real number called a key. The smallest key in a $B_k$ tree can be found quickly, since it is always contained in the root of the tree.

   A binomial queue of size $2^n - 1$ is a group of $n$ $B_k$ trees with the form




$n \geq 1$;

   it therefore has $2^0 + 2^1 + \cdots + 2^{n-2} + 2^{n-1} = 2^n - 1$ nodes, as its name implies. An interesting operation on binomial queues is

   **Program M.** (*Find the minimum*) Given a binomial queue of size $2^n - 1$, we will find $m$, the smallest key in the queue and $B_j$, the tree containing $m$.

| # times executed | |
|---|---|
| 1 | $j \leftarrow 0$; |
| 1 | $m \leftarrow$ smallest key in the $B_0$ tree; |
| 1 | $k \leftarrow 1$; |
| $n$ | **while** $k \leq n$ **do** |
| $n-1$ |     **if** smallest key in the $B_k$ tree is $\leq m$ **then do** |
| $C_n$ |         $j \leftarrow k$; |
| $C_n$ |         $m \leftarrow$ smallest key in $B_k$ tree; |
| |     **end**; |
| $n-1$ |     $k \leftarrow k+1$; |
| | **end**; |

(A) The total running time of this program is $3n+2C_n+1$. The goal of this problem is to find the expected running time, $3n+2E(C_n)+1$. To do that, we need to find $E(C_n)$.

For the analysis, assume that the keys are $1, 2, \ldots, 2^n - 1$, that each of the $(2^n - 1)!$ permutations of the keys are equally likely, and that keys in permutations map onto keys in trees as follows:

$$\begin{array}{ccccc} [X_1] & [X_2 X_3] & \ldots & [X_{2^{n-2}} \ldots X_{2^{n-1}-1}] & [X_{2^{n-1}} \ldots X_{2^n-1}] \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ B_0 & B_1 & & B_{n-2} & B_{n-1} \end{array}$$

The arrangement of the keys within the $B_k$ trees is irrelevant, except that the smallest of the $2^k$ keys in a $B_k$ tree appears in the root, Thus, if $n = 4$ and the permutation is

$$\begin{array}{cccc} [9] & [4\ 3] & [2\ 8\ 1\ 7] & [12\ 13\ 14\ 15\ 6\ 5\ 11\ 10] \\ & \uparrow & \uparrow & \end{array}$$

we have $C_4 = 2$ in **Program M** (the updates of $m$ are shown by the arrows). First show that

$$E(C_n) = \sum_{1 \leq k \leq n-1} \frac{2^k}{2^{k+1} - 1}$$

using the hint given below. Then find the asymptotic value of $E(C_n)$ that equals $f(n) + \alpha + O(1/2^n)$. (The constant $\alpha$ will be in summation form, sum up the first few terms to get $\alpha$ to three decimal places).

**Hint:** Prove that $m$ is updated in the $k^{\text{th}}$ trip through the **while** loop independently of the number of previous updates. Then write $C_n = C_{n,1} + C_{n,2} + \cdots + C_{n,n-1}$, where $C_{n,k} = 1$ if $m$ is updated during the $k^{\text{th}}$ loop, and $C_{n,k} = 0$ otherwise.

(B) **Program M** can be sped up quite a bit if we test the binomial queues in the opposite order. Let **Program M′** be the following modification of **Program M**.

line 1 $j \leftarrow n - 1$;
line 2 $m \leftarrow$ smallest key in the $B_{n-1}$ tree;
line 3 $k \leftarrow n - 2$;
line 4 **do while** $k \geq 0$
line 9 $k \leftarrow k - 1$;

Let $D_n$ be the number of times line 6 and 7 are executed in **Program M′**. The average running time of **Program M′** is $3n + 2E(D_n) + 1$. Show that $E(D_n) = \beta + O(1/2^n)$, where $\beta$ is some constant.

Compute $\beta$ to three decimal places. (The same hint as Part (A) applies.)

What is the asymptotic ratio between the running times of **Programs M′** and **M**? (This tells us how much we saved by modifying **Program M**.)

**End of Homeworks!** If you have reached this point, you can definitely pass this course. Congratulations and have a good vacation!