Workshop 3 Exercise

August 6, 2018

In this week, the following exercises are to be done. The first four exercises can be discussed first and the remaining two are to be coded up during the second hour of the workshop.

1. Order the following functions according to their order of growth (from the lowest to the highest):

$$(n-4)!,\ 2\log_2(n+53)^6,\ 2^{2^n},\ 0.032n^4+2n^3+7,\ \log_e^2 n,\ n^{1/3},\ 3^n$$

2. Indicate whether the first function of each of the following pairs has a smaller, same or larger order of growth (to within a constant multiple) than the second function.

   (a) $100n^2$ and $0.01n^3$

   (b) $\log_2 n$ and $\log_e n$

   (c) $(n-1)!$ and $n!$

3. Determine the efficiency class of the following algorithm:

```
int ABC(int n) {
    int a = 0;
    for(int i=0;i<n;i++) {
       for(int j=0;j<n;j++) {
          a = a + sqrt(i+j+1);
       }
    }
return a;
}
```

4. Given a BST, how would you implement outputing the values in the tree in descending order?

5. Given a BST, write a function which computes the average depth of objects stored in the tree.

6. If all n items to be installed in a BST are available in advance, a balanced tree can be constructed, in which no object has a depth greater than $\log_2(n+1)$. In such a tree, searching is guaranteed to be fast. Write a function that accepts as arguments an array of n objects assumed to be in sorted order, and constructs and returns a balanced binary search tree.