Final Homework

Matt Forbes

December 7, 2010

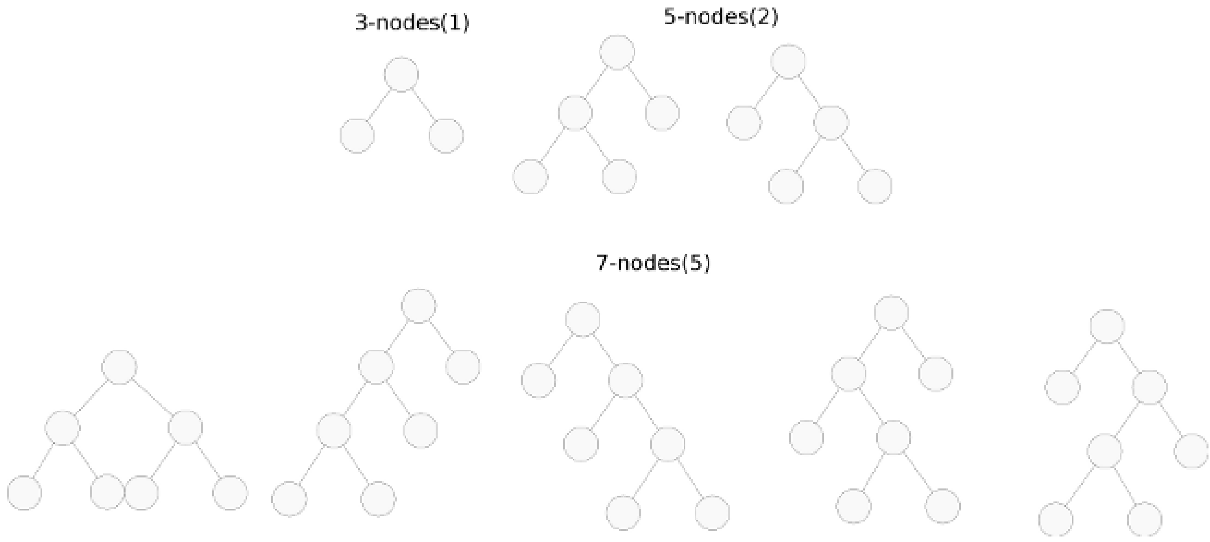
Problem 1

The cost of going from exit j to k is $C_j + C_{j+1} + C_{j+2} + \cdots + C_{k-1}$. I propose the data structure H such that $H_i = C_1 + C_2 + \cdots + C_{i-1}$. To calculate the cost exit j to k using H, it would simply be $H_k - H_j$. This expression expands to $(C_1 + C_2 + \cdots + C_{j-1}) - (C_1 + C_2 + \cdots + C_{k-1})$, which simplifies to $C_j + C_{j+1} + \cdots + C_{k-1}$. Showing that $H_k - H_j$ is equivalent to the cost we calculated for exit j to k. Given that H is already calculated, this computation is a simple subtraction, $O(1)$.

Generating this data structure is very easy and would take $O(n)$ time and holds n elements. Each element H_i is equal to $C_i + H_{i-1}$ which lends itself easily to an accumlating loop from 1 to n.

```
H[1] = C[1]
for i = 2...n:
  H[i] = C[i] + H[i-1]
```

Problem 2



a) $B_3 = 1, B_5 = 2, B_7 = 5$.

b) You can't construct a full binary tree with an even number of nodes. Every node always has zero or two child nodes, meaning everytime the tree grows, it must grow by a multiple of two nodes. So starting with the root, and growing n times, the total number of nodes will always be of the form $1 + 2n$, which is odd.

c)

$$B_n = \left\{ \begin{array}{lr} 1 & n = 1, n = 3 \\ \text{apples} & n > 3 \end{array} \right\}$$