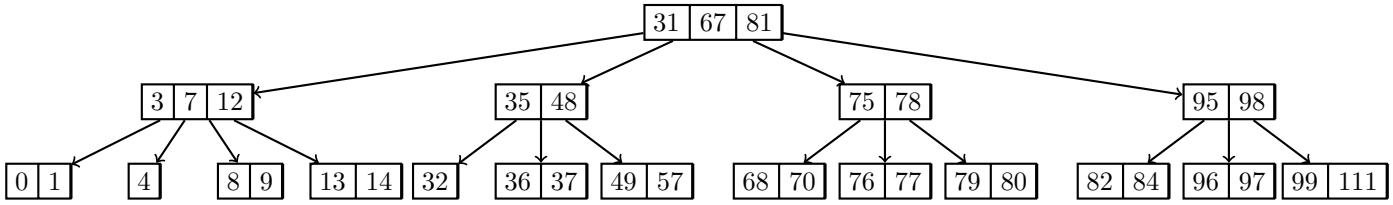
CS6033 Homework Assignment 5*

Due Feb 28th at 5:30 pm
No late assignments accepted

1. (10 points) Write the pseudo code for how to find the minimum key stored in a B-tree and how to find the predecessor of a given key stored in a B-tree. Provide the CPU running time, and the number of disk accesses using Big-Oh notation.

2. (10 points) Insert the following keys: $16, 19, 25, 22, 28, 30$ in this order into the b-tree of degree 2 below using the algorithm discussed in class. Show the tree after each item is inserted.



3. (10 points) Suppose we insert the keys $\{1, 2, \ldots, n\}$ in this order into a an empty B-tree with minimum degree 3. How many nodes does the final B-tree have.

   (5 points extra credit) What is the height of the B-tree?

4. For the following function:

   - (10 points) determine the recurrence relation for the run time using Big-Theta notation
   - (10 points) find the asymptotic closed form solution using the *master method*

   If dividing `X` into `Q` and `R` takes $\Theta(1)$, and `COMBINE` takes $\Theta(n \log(n))$ time.

```
SOME_FUNCTION(X, n)
     if (n = 1) return

        Divide X into Q and R
        A  = SOME_FUNCTION( Q, n/4)
        B = SOME_FUNCTION(R, n/4)
       COMBINE(A, B)
```

*Many of these questions came from outside sources.

5. For each of the following recurrences, give an expression using Big-Theta notation for the runtime, T(n). Assume $T(1) = \Theta(1)$ for all cases.

   (a) (5 points)$T(n) = 3T(n-1) + 2$

   (b) (5 points)$T(n) = 2T(n/2) + 6n - 1$

   (c) (5 points)$T(n) = 2T(n/2) + 6n/log(n)$

   (d) (5 points)$T(n) = 7T(n/2) + \Theta(n^2)$

   (e) (5 points)$T(n) = T(n/2) + \Theta(1)$

   (f) (5 points)$T(n) = 5T(n/4) + \Theta(n^2)$

6. (10 points) Draw the recursion tree for $T(n) = T(n/4) + T(3n/4) + \Theta(n)$ (where $T(1) = \Theta(1)$) and guess a closed form asymptotic solution.

   (10 points) Prove your guess is correct using the substitution method (if your guess is not correct, redo the recursion tree)