CS3343 Analysis of Algorithms Fall 2019
Homework 5
Due 10/25/19 before 11:59pm (Central Time)

1. Hash Table Probabilities (3 points)

(1) (1 point) Suppose 2 keys are inserted into an empty hash table with m slots. Assuming simple uniform hashing, what is the probability of:
  (a) exactly 0 collisions occurring
  (b) exactly 1 collisions occurring

(2) (2 points) Suppose 3 keys are inserted into an empty hash table with m slots. Assuming simple uniform hashing, what is the probability of:
  (a) exactly 0 collisions occurring
  (b) exactly 1 collisions occurring
  (c) exactly 2 collisions occurring

2. Red-Black Trees (2 points)

(1) Company X has created a new variant on red-black trees which also uses blue as a color for the nodes. They call these "red-black-blue trees". Below are the new rules for these trees:

- Every node is red, blue, or black.
- The root is black.
- Every leaf (NIL) is black.
- If a node is red, then both its children are black.
- If a node is blue, then both its children are red or black.
- For each node, all simple paths from the node to descendant leaves contain the same number of black nodes.

  (a) (2 points) In class we found that the height, h, of a red-black tree is $\leq 2\log_2(n+1)$ (where n is the number of keys). Find and prove that a similar bound on height of the red-black-blue trees.

  (**Hint:** You can use the same approach as we did to show $h \leq 2\log_2(n+1)$).

  (b) (0 points - just for fun) Adding an additional color didn't seem to improve our bound on h (i.e., 3 colors allows the tree to become more unbalanced than with 2 colors). What benefit might we get from the extra color?

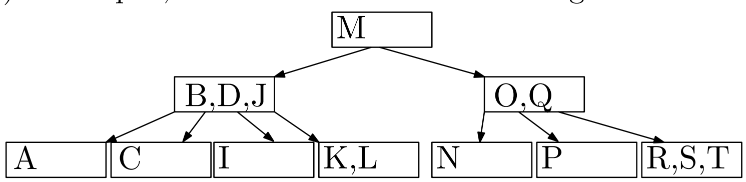
Continued on the back ↪

3. B-trees (4 points)

(1) (2 points) Show the results of inserting the keys

E, F, G, U, V, W, H

in order into the B-tree shown below. Assume this B-tree has minimum degree $k = 2$. Draw only the configurations of the tree just before some node(s) must split, and also draw the final configuration.



(2) (2 points) Suppose you have a B-tree of height h and minimum degree k. What is the largest number of keys that can be stored in such a B-tree? Prove that your answer is correct.

(**Hint:** Your answer should depend on k and h. This is similar to theorem we proved in the B-tree notes).

4. Choose Function (4 points)

Given n and k with $n \geq k \geq 0$, we want to compute the choose function $\binom{n}{k}$ using the following recurrence:

Base Cases: $\binom{n}{0} = 1$ and $\binom{n}{n} = 1$, for $n \geq 0$
Recursive Case: $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$, for $n > k > 0$

(1) (1 point) Compute $\binom{5}{3}$ using repeated calls to the above recurrence.
(2) (2 points) Give pseudo-code for a **bottom-up** dynamic programming algorithm to compute $\binom{n}{k}$ using the above recurrence.
(3) (1 point) Show the dynamic programming table your algorithm creates for $\binom{5}{3}$.

	0	1	2	3
0				
1				
2				
3				
4				
5				