CS 61B Heaps — Exam Prep 12: LLRBs, Hashing, Spring 2020

1 Hashtable Runtimes

Consider a hash table that uses external chaining and also keeps track of the number of keys that it contains. It stores each key at most once; adding a key a second time has no effect. It takes the steps necessary to ensure that the number of keys is always less than or equal to twice the number of buckets (i.e., that the load factor is ≤ 2). Assume that its hash function and comparison of keys take constant time. All bounds should be a function of N, the number of elements in the table. (Fall 2016 MT2)

1. Give $\Theta()$ bounds on the worst-case times of adding an element to the table when the load factor is 1 and when it is exactly 2 before the addition.

   Bound for load factor 1: $\Theta(N)$. Worse case they are all in the same bucket.

   Bound for load factor 2: $\Theta(N)$. Assuming that resize doesn't do an duplicate check. If the resize is implemented such that there is a duplicate check (i.e. resize just calls put), it could be $\Theta(N^2)$.

2. Assume that the hashing function is so good that it always evenly distributes keys among buckets. What now are the bounds on the worst-case time of adding an element?

   Bound for load factor 1: $\Theta(1)$. With a good hash function, you will be bounded by the load factor, which is constant.

   Bound for load factor 2: $\Theta(N)$. Resizing takes linear time.

3. Making no assumption about the goodness of the hashing function, suppose that instead of using linked lists for the buckets, we use some kind of binary search tree that somehow keeps itself "bushy." What bound can you place on the worst-case time for testing to see if an item is in the table?

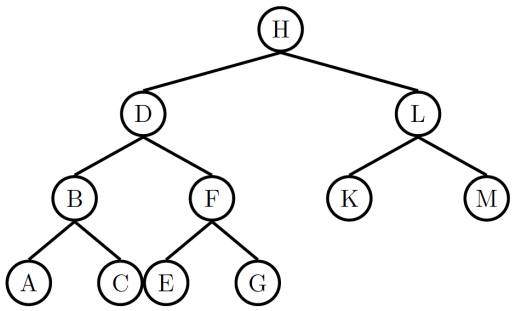
   Bound: $\Theta(\log N)$. Worst case everything hashes to the same bucket, but searching will be $\log N$ because of the bushy tree.

4. Using the same representation as in part (c), but with a very good hash function, as in part (b), what bound can you place on the worst-case time for testing to see if an item is in the table?

   Bound: $\Theta(1)$

2 Min Heaps

Consider the min heap below, where each letter represents some value in the tree. For each question, indicate which letter(s) correspond to the specified value. Assume each value in the tree is unique. (Spring 2018 MT2)



1. Assuming values are inserted into the heap in increasing order, indicate all letters which could represent the following values:

   Smallest value: H
   Median value: K
   Largest value: G

2. Assuming values are inserted into the heap in decreasing order, indicate all letters which could reprsent the following value:

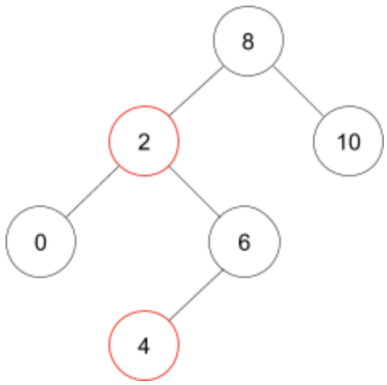
   Smallest value: H
   Median value: L
   Largest value: A

3. Assuming values are inserted into the heap in an unknown order, indicate all letters which could represent the following values:

   Smallest value: H
   Median value: A B C E F G K L M
   Largest value: A C E G K M

3 LLRB Challenge

For this problem, we are working with an LLRB containing the integers 0, 2, 4, 6, 8, 10. Choose an integer x for the below configuration of the LLRB such that the insertion of x into the LLRB requires exactly 7 fixups. A fixup can either be a rotate right, a rotate left, a color flip, or changing the color of the root node. The solution must include the integer x and an enumeration of the fixups in the proper order.



We insert integer 5 into the configuration below and perform the following steps:

1. Rotate 4 left
2. Rotate 6 right
3. Colorflip 5
4. Rotate 2 left
5. Rotate 8 right
6. Colorflip 5
7. Adjust the color of the root node