Homework 5, CSCI 305

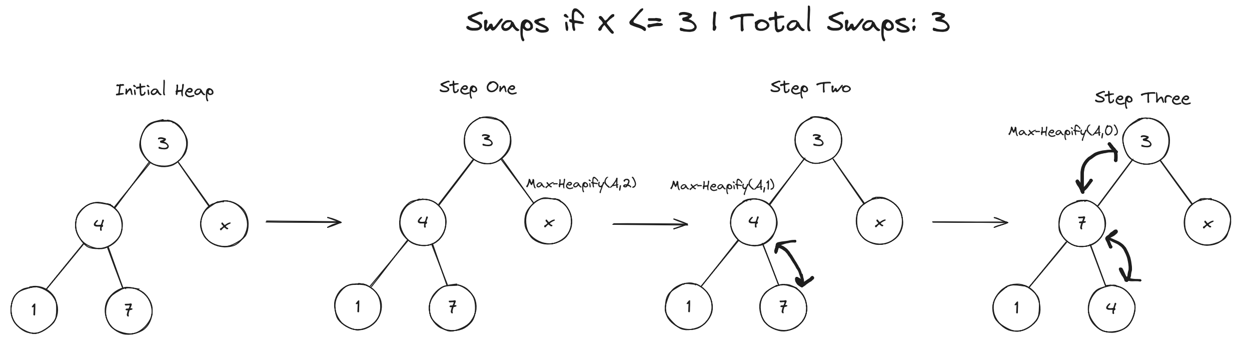
Name: Christopher Farrer
Date: 12/9/23
Professor: K, Harris

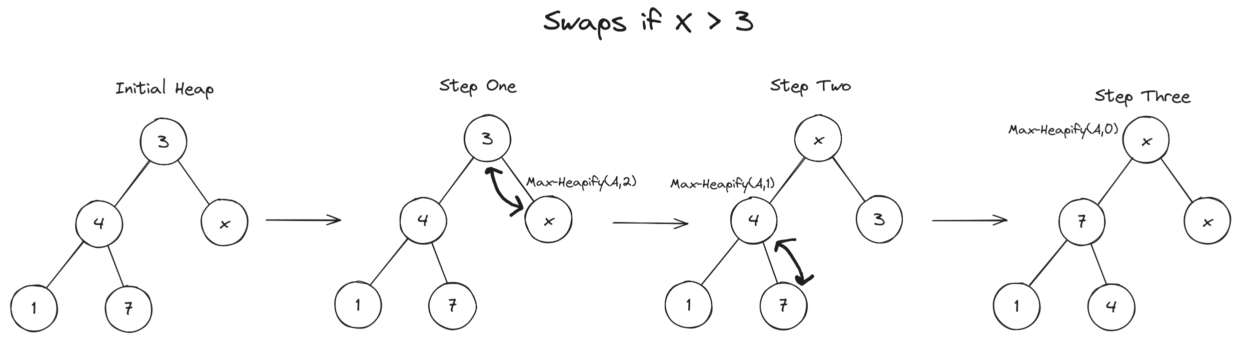
Collaboration

I worked with Ksusha Gotham on this assignment

1. Assume heap A = [3, 4, x, 1, 7] where x is a single digit integer greater than 0. If y is the count of swaps performed during *Build-Max-Heap(A,5)*, then what is/are the value(s) of x such that $y \leq x$. If there are no such values of x, write none.

We need to consider two possible options:

- If $x \leq 3$:
  - 

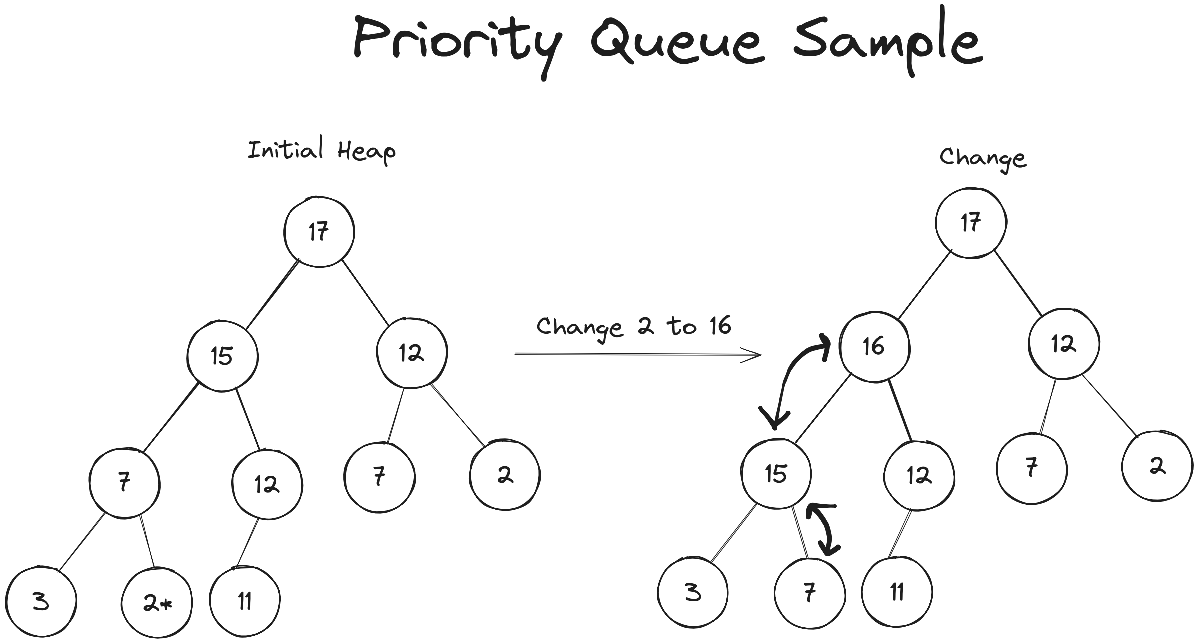
  - To satisfy $y \leq x$, x = 3 would be the only valid option for x.
- If $x > 3$:
  - We reach the following state at step three:
  - 

  - If x < 7, we know only one more swap would be performed, so y = 3 & 3 < x < 7
    - All values of x in this range satisfy $y \leq x$
  - If $x \geq 7$, no more swaps would be required, so y = 1 & $x \geq 7$
    - All values of x in this range satisfy $y \leq x$

Putting everything together, the values of x such that $y \leq x = \{x : x \geq 3\}$

2. Array Q represents a priority queue, implemented as a max-heap. The entries of Q are the priorities of elements, where higher numbers represent higher priorities. Q = [17, 15, 12, 7, 12, 3, 2*, 11]. What change to the priority of the item in the node marked* would require two swaps to re-heapify Q?

Answer:
4. Change 2 to 16



3.Suppose we use a hash function *h* to has *n* distinct keys (we can think of these as any of [n] = {1, 2, ..., n}) into an array T of length m. Assume simple uniform hashing: This means any given elements is equally likely to hash into any of the m slots, independent of where any other element has hashed to. What is the expected number of collisions?

More precisely, define the random variable equal to the number of collisions and compute $\mathbb{E}[X]$. Use indicator random variables to compute the answer.

$$X = |\{(k, l) : k, l \in \{1, \ldots, n\}, k \neq l, \text{ and } h(k) = h(l)\}|$$

Under the assumption of simple uniform hashing, we know:

$$h(k) = i \text{ with probability } \frac{1}{m}$$

Because the probability is uniform for every value, the $\mathbb{E}[X]$ is simply:

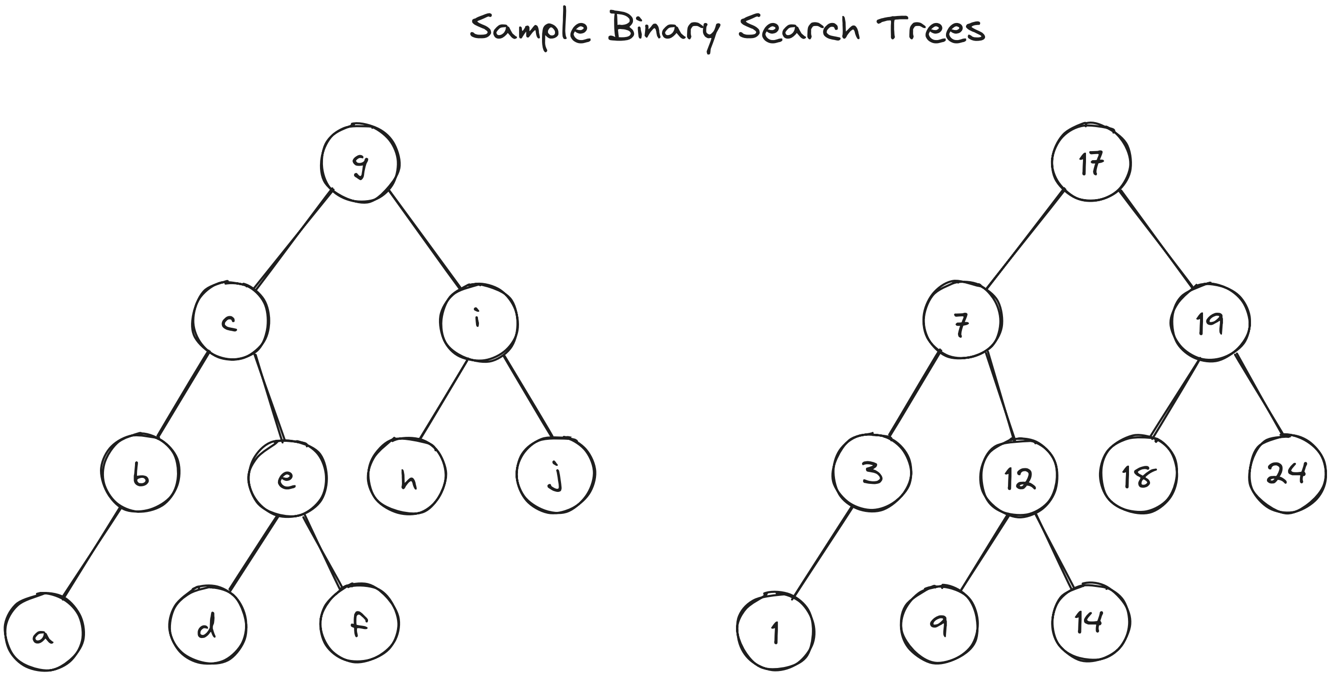
$$E[X] = \sum_{i=1}^{n} \frac{1}{m} = \frac{n}{m}$$

This value is also referred to as the load factor or α

4. In the following binary search tree, finding node *g* requires 1 comparison and finding node *a* requires 4 comparisons. If the BST is queried with a key that doesn't exist in the tree, recall that the search process terminates when NIL is returned as x.left or x.right at the node x. For example, querying with key < a.key would require 5 comparisons.

Suppose X is the random variable equal to the number of comparisons required to find a node chosen at random *from the set of nodes in the tree*, so the null pointer is never returned.

Build a table that answers the following questions about X. Please use exact fractions for probabilities.



i) What is the probability of X = x for all possible values of x?

The max number of comparisons to find a node in the tree is 4. Thus, the question can be restated such that:

$$Pr(X = x) : x \in \{1, 2, 3, 4\}$$

The probability of hitting a node given a certain number of comparisons depends on how many nodes are available at each level (not counting potential NIL values). Thus, we can express the probability for each number of comparisons as the number of possible nodes over the total number of nodes in a piecewise function:

$$Pr(X = x) = \begin{cases} \frac{1}{10} & x = 1 \\ \frac{2}{10} & x = 2 \\ \frac{4}{10} & x = 3 \\ \frac{3}{10} & x = 4 \end{cases}$$

If we take the sum of the probabilities for each x we get 1, which validates the above function.

ii) Calculate $\mathbb{E}[X]$. Show the calculation steps.

The definition of expected value is

$$E[X] = \sum_{x \in S} x \cdot P(X = x)$$

Where $S = \{1, 2, 3, 4\}$

Applying that definition to the probability function defined in the last problem, we get:

$$E[X] = \sum_{x \in S} x \cdot P(X = x) = 1 \cdot \frac{1}{10} + 2 \cdot \frac{2}{10} + 3 \cdot \frac{4}{10} + 4 \cdot \frac{3}{10} = \frac{29}{10} = 2.9$$

Now let's assume that the key which is queried is chosen uniformly from {1, 2,..., 30}. Let Y be the random variable for the number of comparisons required to find the key or return NIL.

Our goal now it to compute $\mathbb{E}[Y]$ under this new sampling. To do this, it will be easier to *condition* on the event that the key is or is not stored in the tree. Let K denote the event "The key is in the tree" and K^c the complement event "the key is not in the tree." We use the notations $Y|K^c$ to mean "Y given that K^c occurred", i.e. "the number of comparisons given that the key is not in the tree."

Useful Formulas & Equations:

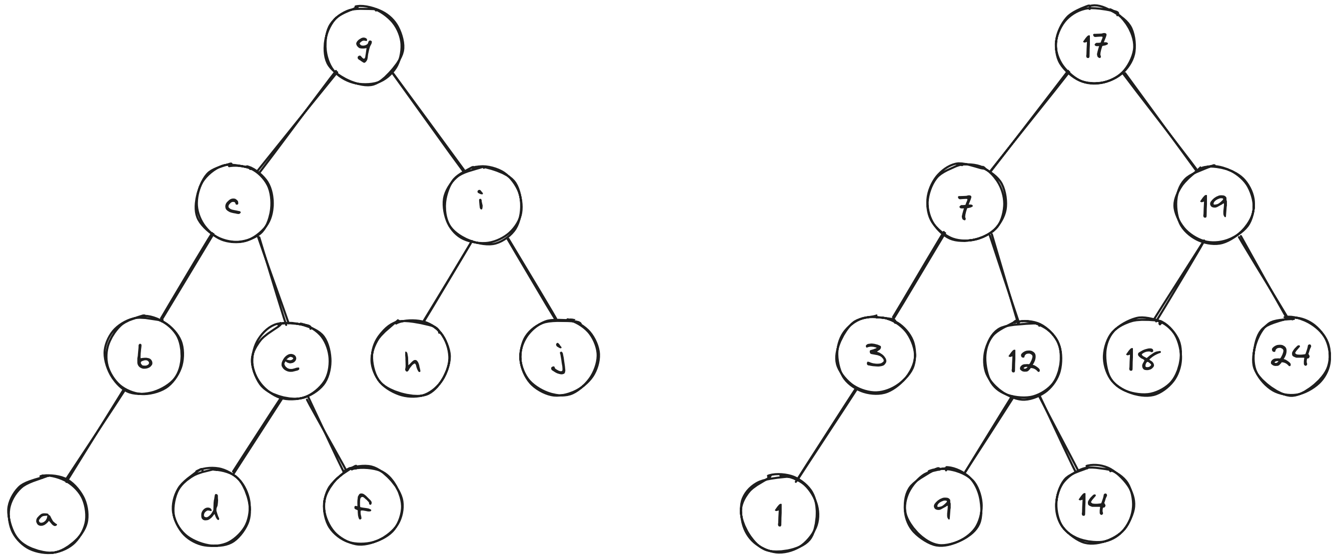
$$\Pr(Y = y) = \Pr(Y = y \text{ and } K) + \Pr(Y = y \text{ and } K^c) =$$

$$\Pr(Y = y|K) \Pr(K) + \Pr(Y = y|K^c) \Pr(K^c)$$

and (often called the law of total expectation):

$$\mathbb{E}[Y] = \mathbb{E}[Y = y|K] \Pr(K) + \mathbb{E}[Y = y|K^c] \Pr(K^c)$$

Sample Binary Search Trees



iii) Calculate $Pr(K), Pr(K^c), \text{ and } Pr(K) + Pr(K^c)$

The universal sample space (U) is {1, 2, 3,... 30} with 30 elements
The sample space of K (S) is {1, 3, 7, 9, 12, 14, 17, 19, 18, 24} with 10 total elements

- $Pr(K) = \frac{10}{30} = \frac{1}{3}$
- $Pr(K^c) = \frac{20}{30} = \frac{2}{3}$
- $Pr(K) + Pr(K^c) = 1$

Also, $Pr(K) = 1 - Pr(K^c)$ and $Pr(K^c) = 1 - Pr(K)$ by definition of the probability of a complement.

iv) Do you have to do any more work to calculate $Pr(Y = y|K)$?

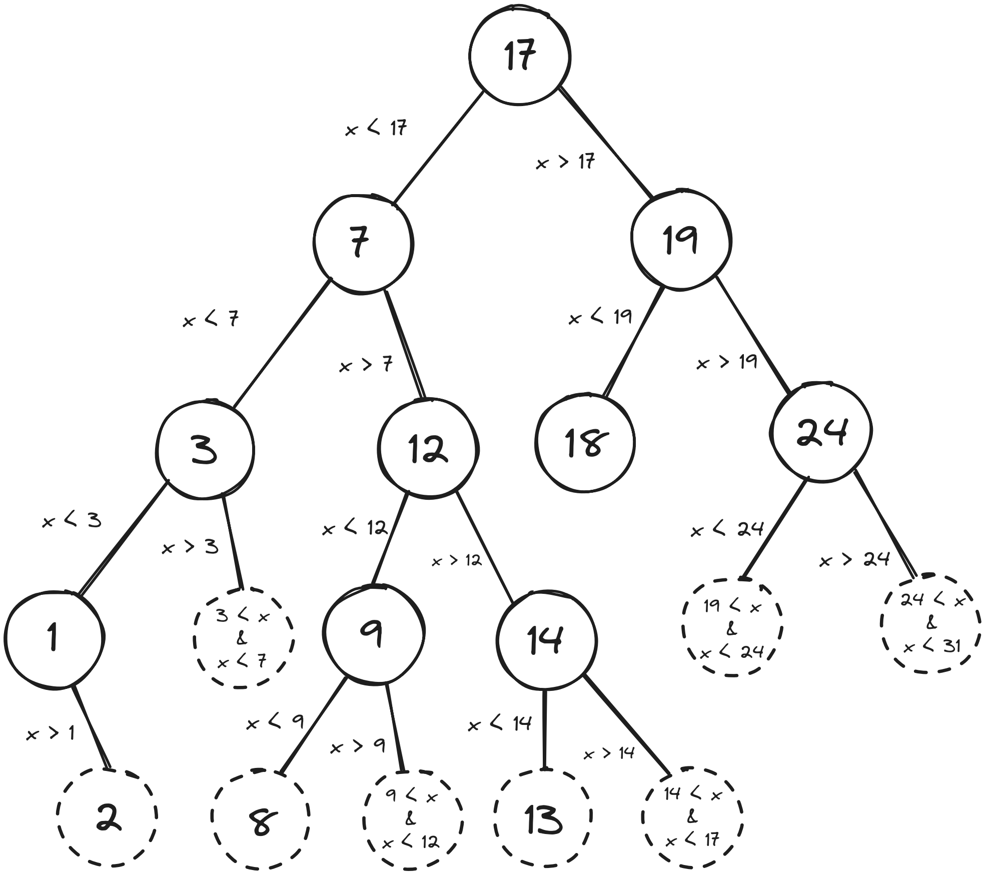
That probability translates to the question: What is the probability it takes a certain amount of comparisons to reach a node given that the key is in the tree?

This is equivalent to $Pr(X = x)$ that we solved in the first part.

v) What is the conditional probability $Pr(Y = y|K^c$ for each possible *y*? Take note that in this event, the tree must be traversed until a node returns NIL for its left or right child.

To solve this question, we must consider the levels in which the elements of K^c would appear in the binary search tree. Using the properties of a BST, we can establish a series of ranges that can appear in each level. Given that levels 1 through 3 are full, we know it must take at least 4 comparisons to return NIL. The graphic below demonstrates these values/ranges.

K Complement Node Positions



Hypothetical nodes are denoted by a dashed border. If a range indicated that only one value would be valid at a position, I input that value rather than a range.

The values that would take 4 comparisons: {4, 5, 6, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30} $n = 13$
The values that would take 5 comparisons: {2, 8, 10, 11, 13, 15, 16} $n = 7$

We can now create a piecewise function in much the same way we solved $Pr(X = x)$.

$$Pr(Y = y|K^c) = \begin{cases} \frac{13}{30} & y = 4 \\ \frac{7}{30} & y = 5 \\ 0 & 1 \leq y \leq 3 \end{cases}$$

vi) What is the conditional expectation $\mathbb{E}[Y|K^c]$?

$$E[X] = \sum_{x \in S} x \cdot P(X = x) = (1 \cdot 0) + (2 \cdot 0) + (3 \cdot 0) + (4 \cdot \frac{13}{30}) + (5 \cdot \frac{7}{30}) = \frac{87}{30} = \frac{29}{10} = 2.9$$

vii) What is the expected value $\mathbb{E}[Y]$?

Using the formula given:

$$\mathbb{E}[Y] = \mathbb{E}[Y = y|K] \Pr(K) + \mathbb{E}[Y = y|K^c] \Pr(K^c)$$

We can plug in our answers to achieve:

$$\mathbb{E}[Y] = \frac{29}{10} \cdot \frac{1}{3} + \frac{29}{10} \cdot \frac{2}{3} = \frac{87}{30} = 2.9$$