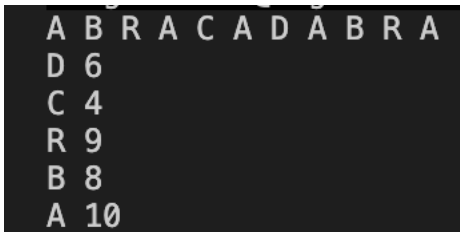
Heimadæmi 6

Magnús Daníel Einarsson

March 2023

Verkefni 1

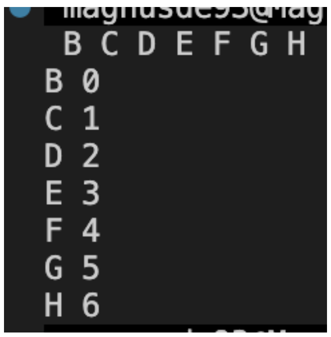
```
public Value get(Key key) {
        if (key == null) throw new IllegalArgumentException("argument to get() is null");
        Node oldfirst = first;
        if (key.equals(first.key)){
            return first.val;
        }
        for (Node x = first; x != null; x = x.next) {
            if (key.equals(x.key)) {
                Value val = x.val;
                first = new Node(x.key, x.val, oldfirst);
                delete(key);
                return val;
            }
        }
        return null;
    }
```



```
A B R A C A D A B R A
D 6
C 4
R 9
B 8
A 10
```

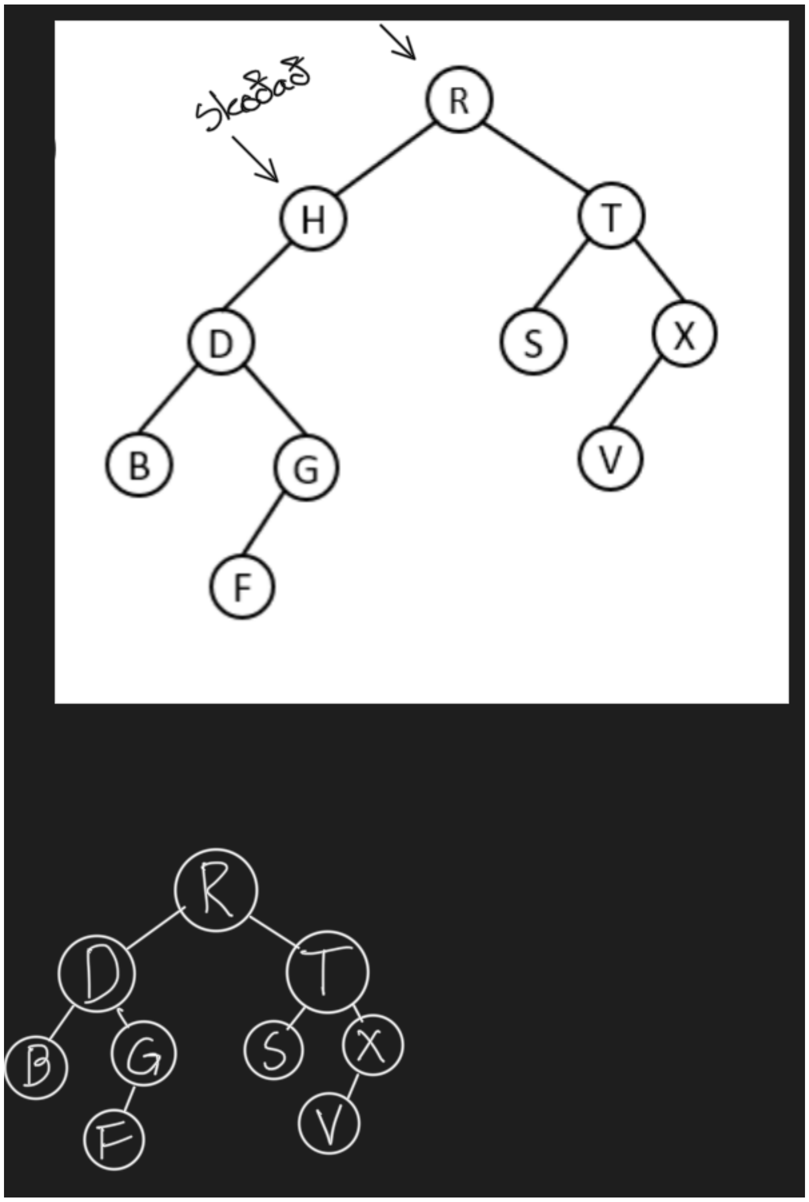
Verkefni 2

```
public void put(Key key, Value val)  {
        if (key == null) throw new IllegalArgumentException("first argument to put() is null");

        if (val == null) {
            delete(key);
            return;
        }

        if (n == 0 || key.compareTo(keys[n-1]) > 0) {
            keys[n] = key;
            vals[n] = val;
            n++;
            return;
        }

        int i = rank(key);

        // key is already in table
        if (i < n && keys[i].compareTo(key) == 0) {
            vals[i] = val;
            return;
        }

        // insert new key-value pair
        if (n == keys.length) resize(2*keys.length);

        for (int j = n; j > i; j--)  {
            keys[j] = keys[j-1];
            vals[j] = vals[j-1];
        }
        keys[i] = key;
        vals[i] = val;
        n++;

        assert check();
    }
```

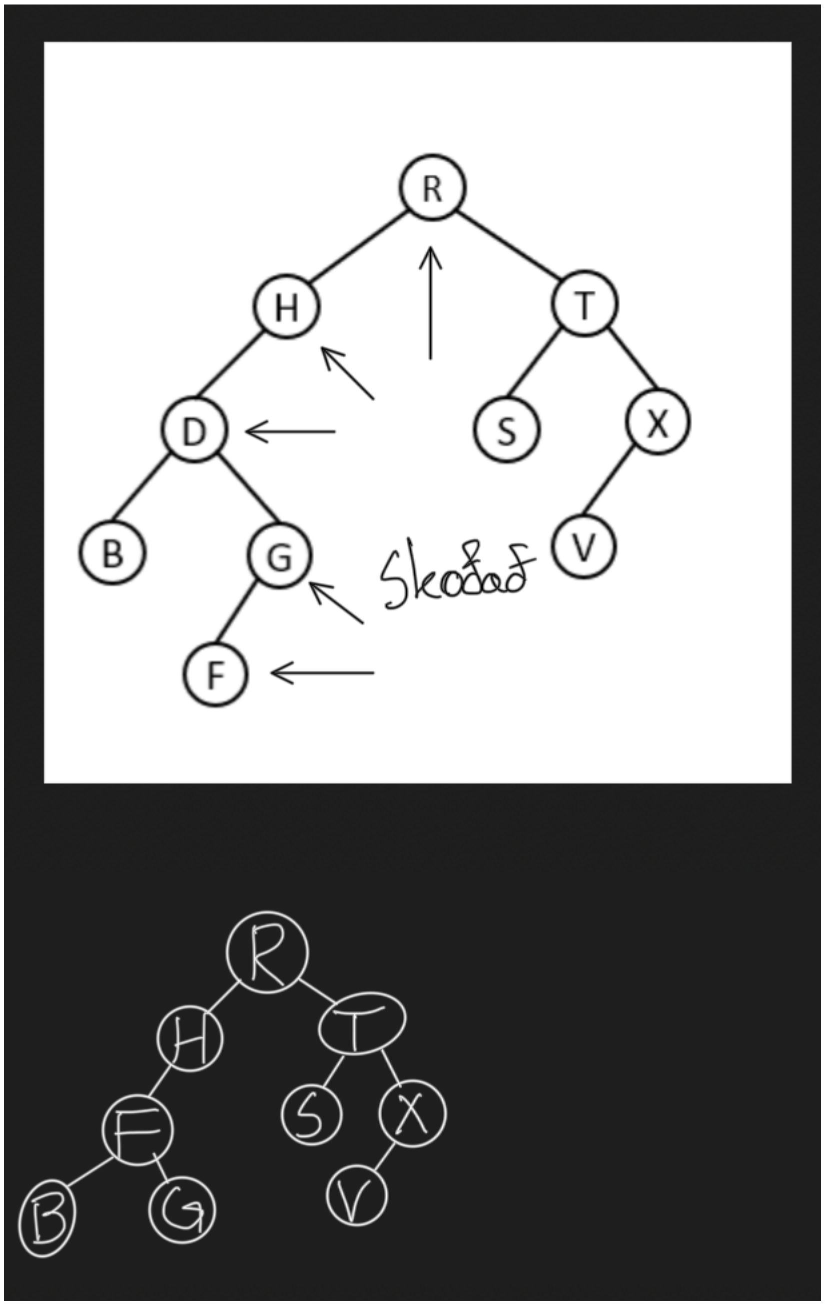


```
  B C D E F G H
B 0
C 1
D 2
E 3
F 4
G 5
H 6
```

Verkefni 3



a)



b)

Verkefni 4

a Við skoðum R, H, D og skilagildið er hnúturinn D.

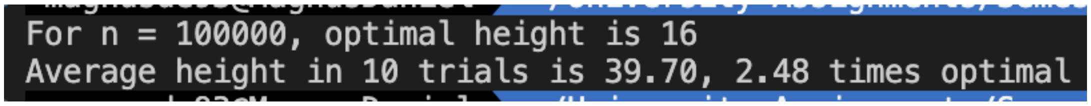
b Við skoðum R, H, T, S og skilagildið er hnúturinn T.

c Við skoðum R, T og skilagildið er lykill hnútarins með gildið 7.

d Við skoðum R, H, D, G, F og skilagildið er hnúturinn D.

Verkefni 5

```java
public class MeasureBST {

    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        int trials = Integer.parseInt(args[1]);
        int sumOptimalHeight = 0;
        int sumHeight = 0;
        for (int t = 0; t < trials; t++) {
            BST<Double, Integer> bst = new BST<>();
            for (int i = 0; i < n; i++) {
                double key = StdRandom.uniform();
                bst.put(key, i);
            }
            sumHeight += bst.height();
            sumOptimalHeight += Math.floor(Math.log(n) / Math.log(2));
        }
        double avgHeight = (double) sumHeight / trials;
        double avgOptimalHeight = (double) sumOptimalHeight / trials;
        double ratio = avgHeight / avgOptimalHeight;
        StdOut.printf("For n = %d, optimal height is %d\n", n, (int) Math.floor(Math.log(n) / Mat
        StdOut.printf("Average height in %d trials is %.2f, %.2f times optimal\n", trials, avgHei
    }
}
```



```
For n = 100000, optimal height is 16
Average height in 10 trials is 39.70, 2.48 times optimal
```