CS222 Homework 3

Stable Matching and Algorithm Analysis

Exercises for Algorithm Design and Analysis by Li Jiang, 2018 Autumn Semester

- 潘佳萌
- 516030910510

1. a. 1st search can find the number in the first layer in the BinarySearch Tree, whose probability is $\frac{1}{n}$
   b. 2nd search can find the number in the second layer in the BinarySearch Tree, whose probability is $\frac{2}{n}$
   c. kth search can find the number in the nth layer in the BinarySearch Tree, whose probability is $\frac{2^{k-1}}{n}$
   $T(n) = \sum_1^k \frac{m*2^{m-1}}{n} = O(log_2 n)$
2.

```
count(num[], lower, upper)
  sums=new[len(num)]
  for n=0:len(num)
    sums[n+1]=sums[n]+num[n]
  return merge(sums, 0, len(sums), lower, upper)
merge(sums[], start, end, lower, upper)
  if r-l<=1
    return 0
  mid=start+(end-start)/2
  result = merge(sums,start,mid,lower,upper)+merge(sums,mid+1,end,lower,upper)
  tmp=new[end-start+1]
  j=k=t=mid+1, i=start, r=0
  for i=0:mid, i++, r++
    while j<=end and sums[i]<=upper
      j++
    while k<=end and sums[i]<lower
      k++
    count+=j-k
    while t <= end and sumss[t] <= sums[i]
      tmp[r++] = sumss[t++];
    tmp[r] = sums[i]
    for i=0:r, ++i
      sums[start + i] = tmp[i]
    return count
```

3.

```
minT(T)
  if T.value < T.left.value or T.value < T.right.value or (T.left and T.right are empty)
    return T.value
  else if T.value > T.left.value and T.value < T.right.value
    return minT(T.right)
  else if T.value < T.left.value and T.value > T.right.value
    return minT(T.left)
  else return min(minT(T.right), minT(T.left))
```