CPTS 233 - HW1

Name: Boxiang Lin

Question 1

Ordered Complexities
2/N
37
sqrt(N)
N
N log (log N)
N log N
N log^2 N
N^1.5
N^2
N^2 log N
N^4
2^(N/2)
2^N

Question 2

1. In O(N) rate of grow to be linearly, so given N = 100 there will be 100/20 much of N for 35 seconds = **100/20 * 35 = 175 seconds**.
2. O(N + log N) can be approximate as O(N), therefore **about the same of 175 seconds would take**.
3. O(N^3), 35s = c*20^3 where c = 35/(20^3), Now N=100 : **100^3 * 35/(20^3) = 4375 seconds.**
4. O(2^N), 35s = c*2^20 where c = 35/(2^20), Now N=100: **2^100 * 35 / (2^20) = 4.23*10^25 seconds**.

Question 3

- A: f() runtime complexity is O(N), g() runtime complexity is O(N).

- B: f() space complexity is O(1) because space for sum variable reusable, g() space complexity is O(N) because new recursive stake until N times.
- C: Return the parameter value as it is.

```
int h (int h){
    return h;
}
```

Question 4

g(n) runtime complexity is O(log N * log N) = O(log^2 N), since f(n) is log N and nested inside of g(n).

Question 5

The while loop take O(N) and the substring() takes O(N - constant) inside the while loop, the whole split method runtime to be O(N^2).

Question 6

```
    public static int kFinder (int n) {
        int [] array = new int [10];
        Arrays.fill(array,0); //using array to store the existence of 0-9
dicimals
                              //value 0 is not exist and value 1 is exist.

        int k = 0;
        while(Arrays.stream(array).sum() < 10)
        //if all the index has 1 then quite the loop, return the current k.
        {
            k++;  //start at 1;
            int product = n*k;
            while(product != 0) { //store the decimal by shifting the number to
the right
                int deci = product % 10;  //getting the last digit
                array[deci] = 1;    //mark last digit to be the index and now
exist
                product = product / 10;  //keep shifting to the right by deivide
the base
            }
            //next product check;
        }
        return k;
    }
```

Because the array size is defined, Arrays.fill and Arrays.stream(array.sum()) could be treated as constant. The outer while loop take k times and the inner while loop product divide 10 each time, so this runtime is O(k*log N).

Question 7

- A: runtime: O(1) space: O(1)

```
public static String oddOrEven(int number){
    return (number % 2 == 0) ? "IsEven" : "IsOdd";
}
```

- B: runtime O(N) space: O(1)

```
public static boolean isContains (List <Integer> list, int value){
    for(int i = 0; i<list.size(); i++){
        if(list.get(i) == value) return true;
    }
    return false;
}
```

- C: runtime O(N) space: O(1)

```
public static int smallest (List <Integer> list){
    int small = list.get(0);
    for(int i = 1; i<list.size(); i++){
        if(list.get(i) < small) small = list.get(i);
    }
    return small;
}
```

- D: runtime O(N^2) space O(1)

```
public static boolean unsortedListCompare(List<Integer> l1, List<Integer>
l1){
    if(l1.size() != l2.size() ) return false;
    for(int i = 0; i<l1.size(); i++){
        for(int j = 0; j<l2.size(); j++){
            if(l1.get(i) != l2.get(j) ) return false;
        }
    }
    return true;
}
```

- E: runtime O(N) space O(1)

```
public static boolean sortedListCompare(List<Integer> l1, List<Integer> l2){
    if(l1.size() != l2.size() ) return false;
    for(int i = 0; i<l1.size(); i++){
        if(l1.get(i) != l2.get(i) )return false;
    }
    return true;
}
```

- F: runtime O(log N), space O(1)

```
public static int indexBST (int val, int [] array){
    int left = -1;  //initialize out of bound
    int right = array.length;
    while(left+1 != right){
        int middle = (left+right)/2;
        if(val<array[middle]) right = middle;
        if(val == array[middle]) return middle;
        if(val > array[middle]) left = middle;
    }
    return -1;
}
```

Question 8

Git is a version control system(software), make the teamwork easier, popularly use in the industry. Git can track the changes we made on a project, reverse back the older version if we wish, and merging teammates codes into the central repository, teammates could therefore working on the individual modules themselves.

Question 9

---

git Clone [HTTPS or SSH]

Question 10

---

git add [file]

git add -A   //for all

Question 11

---

git commit -m "message"

Question 12

---

git push

Question 13

---

git pull

Question 14

```
The String [] args is the command-line argument in a type of String array.
After compiling the .java to .class, we could pass arguments into the parameter
of String [] args by "java className [arguments]".
This is because the JVM will first call the main method, and the arguments after
the java className command is the arguments that pass to the main method
parameter. We could access these argument by args[index].
```