2020-03-01 - Handout – Big O Notation

Q1. LT 567 - Permutation in String

Link: https://leetcode.com/problems/permutation-in-string/

Given two strings **s1** and **s2**, write a function to return true if **s2** contains the permutation of **s1**. In other words, one of the first string's permutations is the **substring** of the second string.

Example 1:
Input: s1 = "ab" s2 = "eidbaooo"

Output: True

Explanation: s2 contains one permutation of s1 ("ba").

Example 2:
Input:s1= "ab" s2 = "eidboaoo"

Output: False

Q2. LT 78 Subset/Power Set

Link: https://leetcode.com/problems/subsets/

Given a set of **distinct** integers, *nums*, return all possible subsets (the power set).

Note: The solution set must not contain duplicate subsets.

Example 1:
Input: nums = [1,2,3]

Output:
```
[
  [3],
 [1],
 [2],
 [1,2,3],
 [1,3],
 [2,3],
 [1,2],
 []
]
```

1

2

What is the running time of the following function?

```
public void Function(int n) {
        int i=1, s=1;
        while( s <= n) {
                i++;
                s= s+i;
                System.out.println("*");
        }
}
```

2

```
public void function(int n){
int i, count=0;
for(int i =1; i*i*i<=n; i++)
        count++;
}
```

3

What is the complexity of the program given below?

```
public void function(int n) {
        int i, j, k , count =0;
        for(i=n/2; i<=n; i++)
                for(j=1; j + n/2<=n; j++)
                        for(k=1; k<=n; k= k * 2)
                                count++;
}
```

4

```
public void function(int n) {
        int i, j, k , count =0;
        for(i=n/2; i<=n; i++)
                for(j=1; j<=n; j= 2 * j)
                        for(k=1; k<=n; k= k * 2)
                                count++;
}
```

5

Find the complexity of the program given below.

```
public  void function( int n ) {
        if(n == 1) return;
        for(int i = 1 ; i <= n ; i + + ) {
                for(int j= 1 ; j <= n ; j + + ) {
                        System.out.println("*" );
                        break;
                }
        }
}
```

6

```
public  void function( int n ) {
        if( n == 1 ) return;
        for(int i = 1 ; i <= n ; i + + )
                for(int j = 1 ; j <= n ; j + + )
                        System.out.println("*" ) ;
        function( n-3 );
}
```

7

```
public void Read(int n) {
        int k = 1;
        while( k < n )
                k = 3k;
}
```

8

```
int Fib(int n)
{
   if (n==0) return 0;
   else if(n>1) return 1;
   else return Fib(n-1)+Fib(n-2);
}
```

9

Running time of following program?

```
public  void function(n) {
        for(int i = 1 ; i <= n ; i + + )
                for(int j = 1 ; j <= n ; j+ = i )
                        System.out.println("*") ;
}
```

10

```
public void function(int n) {
        if(n <= 1) return ;
        for (int i=1 ; i <= 3; i++ )
                f(⌈n/3⌉);
}
```

11

What is the complexity of $\sum_{i=1}^{n} \log i$?

12

```
int recursiveFun1(int n)
{
    if (n <= 0)
        return 1;
    else
        return 1 + recursiveFun1(n-1);
}
```

13

```
int recursiveFun2(int n)
{
    if (n <= 0)
        return 1;
    else
        return 1 + recursiveFun2(n-5);
}
```

14

```
int recursiveFun3(int n)
{
    if (n <= 0)
        return 1;
    else
        return 1 + recursiveFun3(n/5);
}
```

15

```
void recursiveFun4(int n, int m, int o)
{
    if (n <= 0)
    {
        printf("%d, %d\n",m, o);
    }
    else
    {
        recursiveFun4(n-1, m+1, o);
        recursiveFun4(n-1, m, o+1);
    }
}
```

16

```
int recursiveFun5(int n)
{
    for (i = 0; i < n; i += 2) {
        // do something
    }

    if (n <= 0)
        return 1;
    else
        return 1 + recursiveFun5(n-5);
}
```

17

```
public void function(int n) {
        if( n < 2 ) return;
        else    counter = 0;
        for i = 1 to 8 do
                function (n/2);
        for i =1 to n^3 do
                counter = counter + 1;
}
```

18

```
public void function(int n) {
        if(n <= 1) return;
        for (int i=1 ; i <= 3 ; i++ )
                function (n - 1).
}
```

19

Find the complexity of the pseudocode given below:

```
temp  = 1
repeat
        for i = 1 to n
                temp =  temp + 1;
        n  = n/2;
until n <= 1
```

20

Running time of the following program?

```
publicvoid  function(int n) {
        for(int i = 1 ; i <= n ; i + + )
                for(int j = 1 ; j <= n ; j * = 2 )
                        System.out.println("*");
}
```

21

Find the complexity of the below function:

```
public void function(int n) {
        int i=1;
        while (i < n) {
                int j=n;
                while(j > 0)
                        j = j/2;
                i=2*i;
        } // i
}
```

22

Find the complexity of the below function:

```
public void function(int n) {
   for (int i = 0; i<n; i++)
      for(int j=i; j<i*i; j++)
         if (j %i == 0){
            for (int k = 0; k < j; k++)
               printf(" * ");
         }
}
```

23

Running time of the following program?

```
public void function(int n) {
        for(int i = 1 ; i <= n/3 ; i + + )
                for(int j = 1 ; j <= n ; j += 4 )
                        System.out.println(" * ");
}
```

24

```
public void function(int n) {
    if(n <=  1) return;
    if(n > 1) {
            System.out.println(" * ");
            function(n/2);
            function(n/2);
        }
}
```

references – data structures and algorithms made easy in java by Narasimha Karumanchi

– Stack overflow

Food for thought

1

```
public void function (int n) {
        if(n <= 1)  return;
        for(int i = 1; i < n; i + +)
                System.out.println("*");
        function ( 0.8n ) ;
}
```

2

```
public int function (int n) {
        if(n <= 2) return 1;
        else
                return (Function (floor(sqrt(n))) + 1);
}
```

3

```
public int gcd(n,m){
        if (n%m ==0) return m;
        n = n%m;
        return gcd(m,n);
}
```

4) Given a target find all the subsets which sum to K with same number being used more than once.