Data Structures and Algo in Java - Day 33

import java.util.\*;

public class day33

{

public static void main (String [] args)

{

String s = "tree";

sortCharactersByFrequency(s);

String s1 = "(1+(2\*3)+((8)/4))+1";

nestingParanthesesDepth(s1);

}

public static void sortCharactersByFrequency(String s)

{

// step 1 to map the letters to frequency

HashMap<Character,Integer> map = new HashMap<>();

for(char ch: s.toCharArray())

{

map.put(ch,map.getOrDefault(ch, 0)+1);

}

// this creates a map of t->1 r->1 e->2 now lets but the 1's into a bucket

List<Character>[] Bucket = new ArrayList[s.length()+1];

for(char key:map.keySet())

{

int freq = map.get(key);// key = t which is 1

if(Bucket[freq]==null) //[1] == 0 . so [1] = [t] ; [1] != 0 . so we just add [1] = [t,r] . [2] == null . [2] = [e]

{

Bucket[freq] = new ArrayList<>();

}

Bucket[freq].add(key);

}

StringBuilder result = new StringBuilder();

for(int i=Bucket.length-1;i>=0;i--)

{

if(Bucket[i]!=null)

{

for(char c:Bucket[i])

{

for(int j=0;j<i;j++)

{

result.append(c);

}

}

}

}

System.out.println(result);

}

public static void nestingParanthesesDepth(String s)

{

int count = 0;

int maxCount = 0;

for(char ch:s.toCharArray())

{

if(ch=='(')

{

count++;

maxCount = Math.max(count, maxCount);

}

if(ch==')')

{

count--;

}

}

System.out.println(maxCount);

}

}