Remove outermost Paranthesis

18 October 2024 22:36

```java
class Solution {
    public String removeOuterParentheses(String s)
    {
        int n=s.length(); //we use length()in string
        String ans="";
        int cnt1=0;

        for(int i=0;i<n;i++)
        {
            if(s.charAt(i)=='(')
            {
                cnt1++;
                if(cnt1>1)
                {
                    ans+=s.charAt(i);
                }
            }
            else if(s.charAt(i)==')')
            {
                cnt1--;
                if(cnt1>0)
                {
                    ans+=s.charAt(i);
                }
            }

        }
        return ans;

    }
}
```

Reverse words in a given string / Palindrome Check

19 October 2024 00:08

#GIVING TLE WHY??

```
class Solution {
    public String reverseWords(String s)
    {
        int n=s.length();
        String result=new String();
        int i=0;
        while(i<n&&(s.charAt(i)==' '))
        {
            i++;
        }
        while(n-1>=0&&(s.charAt(n-1)==' '))
        {
            n--;
        }
        while(i<n)
        {
            if(s.charAt(i)>='a'&&s.charAt(i)<='z')
            {
                String ans="";
                while(i<n&&(s.charAt(i)>='a'&&s.charAt(i)<='z'))
                {
                    ans+=s.charAt(i);
                    i++;
                }
                result=ans+result;
            }
            else if(s.charAt(i)==' ')
            {
                while(i<n&&(s.charAt(i)==' '))
                {
                    i++;
                }
                result=' '+result;
            }
        }

        return result;

    }
}
```

#CORRECT CODE A:

```
class Solution {
    public String reverseWords(String s)
    {
        int n=s.length();
        String result=new String();
        int i=0;
        while(i<n&&(s.charAt(i)==' '))
        {
            i++;
        }
        while(n-1>=0&&(s.charAt(n-1)==' '))
```

```
        {
            n--;
        }
        while(i<n)
        {
            if(s.charAt(i)!=' ')
            {
                String ans="";
                while(i<n&&(s.charAt(i)!=' '))
                {
                    ans+=s.charAt(i);
                    i++;
                }
                result=ans+result;
            }
            else
            {
                while(i<n&&(s.charAt(i)==' '))
                {
                    i++;
                }
                result=' '+result;
            }
        }

        return result;

    }
}
```

#CORRECT CODE B:

```
class Solution {
    public String reverseWords(String s)
    {
        int n=s.length();
        StringBuilder result=new StringBuilder();
        int i=0;
        while(i<n&&(s.charAt(i)==' '))
        {
            i++;
        }
        while(n-1>=0&&(s.charAt(n-1)==' '))
        {
            n--;
        }
        while(i<n)
        {
            if(s.charAt(i)!=' ')
            {
                StringBuilder ans=new StringBuilder();
                while(i<n&&(s.charAt(i)!=' '))
                {
                    ans.append(s.charAt(i));
                    i++;
                }
                if(result.length()>0)
                {
                    result.insert(0,' ');
                }
                result.insert(0,ans);
                ans.setLength(0);
            }
            else
            {
                while(i<n&&(s.charAt(i)==' '))
```

```
                {
                    i++;
                }
            }
        }

        return result.toString();

    }
}
```

Largest odd number in a string

19 October 2024 00:08

```java
class Solution {
    public String largestOddNumber(String num)
    {
        int n=num.length();
        int i=n-1;
        while(i>=0)
        {
            if((num.charAt(i)-'0')%2!=0)
            {
                break;
            }
            i--;
        }
        String s=new String();
        s=num.substring(0,i+1);
        return s;
    }
}
```

Longest Common Prefix

19 October 2024 00:08

#Easy solution

```
class Solution {
    public String longestCommonPrefix(String[] strs)
    {
        int n=strs.length;
        if(n==1)
        {
            return strs[0];
        }
        String a=strs[0];
        String b=strs[1];
        int i=0;
        int j=0;
        String ans=new String();
        while(i<a.length()&&j<b.length())
        {
            if(a.charAt(i)==b.charAt(i))
            {
                ans+=a.charAt(i);
                i++;
                j++;
            }
            else
            {
                break;
            }
        }

        for(i=2;i<n;i++)
        {
            a=ans; //flow
            b=strs[i];
            int k=0;
            int l=0;
            String res=new String();
            while(k<a.length()&&l<b.length())
            {
                //String res=new String();
                if(a.charAt(k)==b.charAt(l))
                {
                    res+=a.charAt(k);
                    k++;
                    l++;
                }
                else
                {

                    break;
                }
            }
            ans=res;
        }
        return ans;
    }
}
```

Isomorphic String***

19 October 2024 00:09

check whether one string is a rotation of another

19 October 2024 00:09

```java
class Solution {
    public boolean rotateString(String s, String goal)
    {
        if(s.length()!=goal.length())
        {
            return false;
        }
        String res=s+s;
        if(res.contains(goal))     //Contain function
        {
            return true;
        }
        return false;
    }
}
```

Check if two strings are anagram of each other

19 October 2024 00:09

```
class Solution {
    public boolean isAnagram(String s, String t)
    {
        if(s.length()!=t.length())
        {
            return false;
        }
        Map<Character,Integer>mp=new HashMap<>();
        for(int i=0;i<s.length();i++)
        {
            if(mp.containsKey(s.charAt(i)))
            {
                mp.put(s.charAt(i),mp.get(s.charAt(i))+1);
            }
            else
            {
                mp.put(s.charAt(i),1);
            }
        }
        // traversing string 2
        for(int i=0;i<t.length();i++)
        {
            if(mp.containsKey(t.charAt(i)))
            {
                mp.put(t.charAt(i),mp.get(t.charAt(i))-1);
                if(mp.get(t.charAt(i))==0)
                {
                    mp.remove(t.charAt(i));
                }

            }
            else
            {
                return false;
            }
        }
        return true;
    }
}
```

Sort Characters by frequency***

19 October 2024 00:09

```
class Solution {
    class pair{
        int a;
        char b;
        pair(int a,char b)
        {
            this.a=a;
            this.b=b;
        }
    }
    public String frequencySort(String s)
    {
        HashMap<Character,Integer>mp=new HashMap<>();
        for(int i=0;i<s.length();i++)
        {
            if(mp.containsKey(s.charAt(i)))
            {
                mp.put(s.charAt(i),mp.get(s.charAt(i))+1);
            }
            else
            {
                mp.put(s.charAt(i),1);
            }
        }
        PriorityQueue<pair> pq = new PriorityQueue<>(new Comparator<pair>()
        {
            @Override
            public int compare(pair p1,pair p2)
            {
                return Integer.compare(p2.a,p1.a);
            }
        });
        // for (Map.Entry entry : mp.entrySet())
        // {
        //     pq.add(new pair(entry.getValue(),entry.getKey()));

        // }
        for (Map.Entry<Character, Integer> entry : mp.entrySet()) {
            pq.add(new pair(entry.getValue(), entry.getKey())); // Add pair (count, character)
        }
        String ans="";
        while(pq.size()>0)
        {
            pair p=pq.poll();
            int cnt=p.a;
            char c=p.b;
            while(cnt>0)
            {
                ans+=c;
                cnt--;
            }
        }
        return ans;
    }
}
```

Maximum Nesting Depth of Paranthesis

19 October 2024 00:09

Roman Number to Integer and vice versa

19 October 2024 00:09

Implement Atoi

19 October 2024 00:09

Count Number of Substrings

19 October 2024 00:09

Longest Palindromic Substring[Do it without DP]

19 October 2024 00:09

Sum of Beauty of all substring

19 October 2024 00:09

Reverse Every Word in A String

19 October 2024 00:10

```
class Solution {
    public String reverseWords(String s)
    {
        int n=s.length();
        String result=new String();
        int i=0;
        while(i<n&&(s.charAt(i)==' '))
        {
            i++;
        }
        while(n-1>=0&&(s.charAt(n-1)==' '))
        {
            n--;
        }
        while(i<n)
        {
            if(s.charAt(i)!=' ')
            {
                String ans="";
                while(i<n&&(s.charAt(i)!=' '))
                {
                    ans+=s.charAt(i);
                    i++;
                }
                result=ans+result;
            }
            else
            {
                while(i<n&&(s.charAt(i)==' '))
                {
                    i++;
                }
                result=' '+result;
            }
        }

        return result;

    }
}
```