牛客网求职算法

直播答疑

牛客网
NOWCODER

简介

牛客网2020最新求职算法——真题精讲高级班
面向BAT、字节跳动等高难度公司，详细讲解40道左右不同类型最新的笔试面试算法真题，并提供最优解和代码，搭配课后作业强化训练。

牛客网：一个提供海量校招真题及专项练习题，笔经面经，招聘信息，学习资源及交流的平台。求职之前，先上牛客https://www.nowcoder.com/

笔经面经

学习交流

合并区间

用x,y表示一个整数范围区间，现在输入一组这样的范围区间(用空格隔开)，请输出这些区间的合并。

输入一行整数，多个区间用空格隔开。区间的逗号是英文字符。
输出合并后的区间，用过空格隔开

输入
1,3 2,5
输出
1,5

输入
1,3 2,5 8,10 11,15
输出
1,5 8,10 11,15

备注:
x,y均为正整数，并且x<=y。

合并区间

用x,y表示一个整数范围区间，现在输入一组这样的范围区间(用空格隔开)，请输出这些区间的合并。

输入一行整数，多个区间用空格隔开。区间的逗号是英文字符。
输出合并后的区间，用过空格隔开

输入
1,3 2,5
输出
1,5

输入
1,3 2,5 8,10 11,15
输出
1,5 8,10 11,15

备注:
x,y均为正整数，并且x<=y。

什么样的两个区间可以合并?

一个区间的起始点在另一个区间之间

如何合并?

合并区间

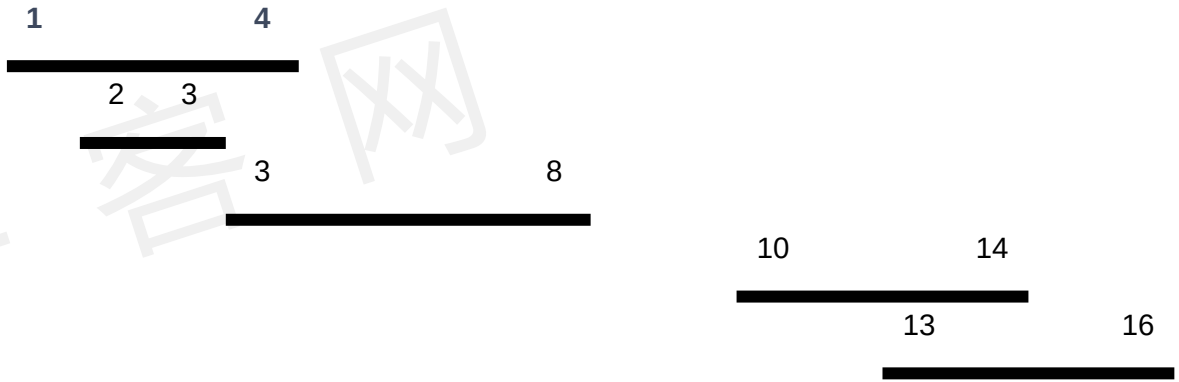
用x,y表示一个整数范围区间，现在输入一组这样的范围区间(用空格隔开)，请输出这些区间的合并。

输入一行整数，多个区间用空格隔开。区间的逗号是英文字符。
输出合并后的区间，用过空格隔开

输入
1,3 2,5
输出
1,5

输入
1,3 2,5 8,10 11,15
输出
1,5 8,10 11,15

[1,4]
[2,3]
[3,8]
[10,14]
[13,16]



备注:
x,y均为正整数，并且x<=y。

合并区间

用x,y表示一个整数范围区间，现在输入一组这样的范围区间(用空格隔开)，请输出这些区间的合并。

输入一行整数，多个区间用空格隔开。区间的逗号是英文字符。
输出合并后的区间，用过空格隔开

输入
1,3 2,5
输出
1,5

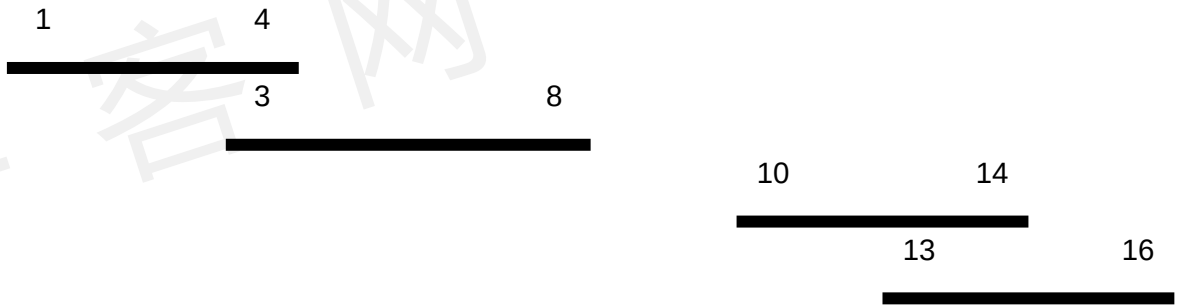
输入
1,3 2,5 8,10 11,15
输出
1,5 8,10 11,15

[1,4]

[3,8]

[10,14]

[13,16]



备注:
x,y均为正整数，并且x<=y。

合并区间

用x,y表示一个整数范围区间，现在输入一组这样的范围区间(用空格隔开)，请输出这些区间的合并。

输入一行整数，多个区间用空格隔开。区间的逗号是英文字符。
输出合并后的区间，用过空格隔开

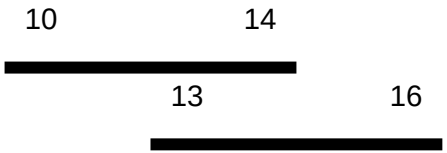
输入
1,3 2,5
输出
1,5

输入
1,3 2,5 8,10 11,15
输出
1,5 8,10 11,15

[1,8]

[10,14]

[13,16]



备注:
x,y均为正整数，并且x<=y。

合并区间

用x,y表示一个整数范围区间，现在输入一组这样的范围区间(用空格隔开)，请输出这些区间的合并。

输入一行整数，多个区间用空格隔开。区间的逗号是英文字符。
输出合并后的区间，用过空格隔开

输入
1,3 2,5
输出
1,5

输入
1,3 2,5 8,10 11,15
输出
1,5 8,10 11,15

[1,8]

[10,16]

备注:
x,y均为正整数，并且x<=y。

合并区间-代码

```
#include<iostream>
#include<algorithm>
using namespace std;
const int MAXN = 1e5+10;
struct node{
    int s,e;
}Interval[MAXN];
bool visit[MAXN];
bool cmp(node a,node b)
{
    if(a.s==b.s){ return a. e< b.e;}
    return a.s < b.s;
}
```

```
int main()
{
    int a,b;
    char c;
    int cnt = 0;
    while(cin >> Interval[cnt].s >> c >> Interval[cnt].e)
    {
         visit[cnt]= true;
         cnt++;
    }
    sort(Interval,Interval+cnt,cmp);
    for(int i = 1 ; i < cnt ; i++)
    {
        if(Interval[i].s <= Interval[i-1].e)//合并
        {
            Interval[i].s = min(Interval[i-1].s,Interval[i].s);
            Interval[i].e = max(Interval[i-1].e,Interval[i].e);
            visit[i-1]=false;//合并后只保留一个区间，前一个区间不要了
        }
    }
    for(int i = 0 ; i < cnt ; i++){
        if(visit[i]){
          cout<<Interval[i].s<<","<<Interval[i].e<<" ";}
    }
    return 0;
}
```

括号字符串的最长有效长度

给定一个括号字符串str，返回最长的能够完全正确匹配括号字符字串的长度。

输入一行字符串，代表str。
输出一个整数，代表括号字符串的最长有效长度。

输入
(()())
输出
6

输入
())
输出
2

括号字符串的最长有效长度

给定一个括号字符串str，返回最长的能够完全正确匹配括号字符字串的长度。

输入一行字符串，代表str。
输出一个整数，代表括号字符串的最长有效长度。

输入
(()())
输出
6

输入
())
输出
2

以某个位置开始，最长的有效字符串的构成：

”(”+一串有效的括号字符串+”)”+一串有效的括号字符串

例：(()())(()())

设**dp[i]**表示以**i**位置开始的最长有效括号字符串的长度

括号字符串的最长有效长度-dp 代码

给定一个括号字符串str，返回最长的能够完全正确匹配括号字符字串的长度。

输入一行字符串，代表str。
输出一个整数，代表括号字符串的最长有效长度。

输入
(()())
输出
6

输入
())
输出
2

```
#include<iostream>
using namespace std;
const int MAXN = 1e5+10;
int main()
{
    string s;
    cin >> s;
    int ans = 0;
    int dp[MAXN]={0};
    for(int i = s.length()-2 ; i >= 0 ; i--)
    {
        if(s[i]=='(')
        {
            int next = i+dp[i+1]+1;//去掉左括号+一段有效的括号字符串之后的位置
            if(next <s.length() && s[next]==')')//+')'
            {
                dp[i] = dp[i+1]+2;
                if(next+1 <s.length()){dp[i]+=dp[next+1];}//加后面的一段有效的括号字符串
            }
        }
        ans = max(ans,dp[i]);
    }
    cout<<ans<<endl;
    return 0;
}
```

括号字符串的最长有效长度-栈 代码

给定一个括号字符串str，返回最长的能够完全正确匹配括号字符字串的长度。

输入一行字符串，代表str。
输出一个整数，代表括号字符串的最长有效长度。

输入
(()())
输出
6

输入
())
输出
2

```
#include<iostream>
#include<stack>
using namespace std;
struct node{
    char c;
    int index;
};
const int MAXN= 1e5+10;
int main()
{
    string s;
    cin >> s;
    stack<node> S;
    int ans = 0;
    for(int i = 0 ;i < s.length();i++)
    {
        if(s[i]==')'&&!S.empty()&&S.top().c=='(')
        {
                S.pop();
                int b = 0;
                if(!S.empty()){ans = max(ans,i-S.top().index);}
                else{ans = i;}
        }
        else{S.push(node{s[i],i});}
    }
    cout<<ans<<endl;
    return 0;
}
```

THANK YOU

查看更多笔经面经

牛客网
NOWCODER