2022级HAUT新生周赛 (二) @武其轩专场题解

104 浏览 0 回复 2022-11-05



Kesuan_Excelsior



默认分享链接

https://acm.haut.edu.cn/contest.php?cid=1080

本题解并非按照周赛题目顺序撰写,请按照题目名称辨别。

A.喂喂, 你知道魔法少女的这个传闻吗?

纯签到题,可以直接用程序计算并输出。

```
#include <bits/stdc++.h>
using namespace std;

int main()

cout << 4 * (11 * 4514 + 114 * 514 + 1145 * 14) << endl;

return 0;
}</pre>
```

B.这就是绝交证明书

本题数据量较大, 如果暴力模拟会导致时间超限。

所以使用一个比较简单的算法来处理——前缀和。

以题解代码为例:数组cnt中cnt[i](1 <= i <= n)表示为第1到第i级台阶共写有的名字个数。

则 cnt[r] - cnt[l-1] 即表示为第l到第r级台阶共写有的名字个数。

每次询问只需要讲行一次减法运算即可,相比遍历求和大大减少了时间复杂度。

有兴趣的同学可以了解一下: https://blog.csdn.net/qq 52156445/article/details/115839679

```
#include <stdio.h>
1
2
    int cnt[100010] = {0};
3
4
5
    int main()
6
7
       int n;
       scanf("%d", &n);
8
       for (int i = 1; i <= n; i++)</pre>
9
10
```

```
}
14
15
       int t;
16
       scanf("%d", &t);
17
       while (t--)
18
19
       {
20
         int 1, r;
         scanf("%d%d", &1, &r);
21
         printf("%d\n", cnt[r] - cnt[l - 1]);
22
23
24
25
       return 0;
26
```

C.我们之间没有你插手的余地哦?

多实例的字符串题目,需要选择合适的读入方式,思路题(方法不唯一)。 需要说明的是,这些情况不表示为两个人之间存在且只存在空隙: "a11a", "aa1 0 1aa"。 这里通过定义一个flag旗帜,flag不同的值代表不同的状态。

- 当遍历找到一个'1'时,记录下当前坐标后继续遍历后面的位置(即代码中的内层for循环)如果 其后一个位置是空隙则使flag为1。
- 如果内层for循环找到第二个'1'时flag为1,则使flag为2,如果flag不为1(0)则退出该循环。
- 如果内层for循环找到非空隙且非'1'字符则退出循环。 最后通过flag的值判断输出Y/N即可。

这里分享一个来自lcl学长(%%%)的解法,很有意思,感兴趣的同学可以看看: https://pastebin.ubuntu.com/p/TsQcR3wY2t/

```
#include <stdio.h>
 1
 2
     #include <string.h>
     int a[100010] = \{0\};
 3
 4
     int main()
 5
     {
         int t;
 6
 7
         scanf("%d", &t);
 8
         while (t--)
 9
10
         {
11
            int n;
            scanf("%d", &n);
12
           getchar();
13
14
           char s[110];
15
           gets(s);
16
           int len = strlen(s);
           int flag = 0;
17
           for (int i = 0; i < len; i++)</pre>
18
19
20
              flag = 0;
              if (s[i] == '1')
21
```

```
if (s[j] == ' ')
25
26
                     flag = 1;
27
                   else if (s[j] == '1')
28
29
                     if (flag == 1)
30
                     {
31
                       flag = 2;
32
                       break;
33
34
                   }
35
                   else
36
                     break;
37
                 }
                if (flag == 2)
38
39
                   printf("Y\n");
40
41
                   break;
42
                 }
              }
43
44
            }
            if (!flag)
45
46
              printf("N\n");
47
          }
48
49
          return 0;
50
     }
```

D.我什么都愿意做

直接按照活动的收益逆序排序然后求出前n个收益的和即可。 感兴趣的同学可以去了解一下不同的排序方法。

PS: 因为数据大小问题需要使用 long (长) long (长)。

```
1
     #include <stdio.h>
 2
 3
     long long a[10010] = {0};
 4
 5
     int main()
 6
 7
       int n;
       scanf("%d", &n);
 8
 9
10
       int t;
       scanf("%d", &t);
11
12
       for (int i = 0; i < t; i++)</pre>
13
         scanf("%11d", &a[i]);
14
       for (int i = 0; i < t - 1; i++)</pre>
15
         for (int j = i + 1; j < t; j++)
16
           if (a[i] < a[j])
17
              long long t = a[i];
18
              a[i] = a[j];
19
```

Excelsion Excelsion

```
for (int i = 0; i < n; i++)
ans += a[i];

printf("%1ld\n", ans);

return 0;
}</pre>
```

E.绝对不能回邮件哦

判断字符然后输出对应的字符串即可。

需要注意的是输入字符串会包含空格,以及输出对空格要单独处理。

```
#include <stdio.h>
 1
 2
     #include <string.h>
 3
 4
     char a[27][5] = {
          "01",
 5
 6
          "1000",
 7
          "1010",
          "100",
 8
 9
          "0",
          "0010",
10
          "110",
11
12
          "0000",
13
          "00",
          "0111",
14
          "101",
15
          "0100",
16
          "11",
17
18
          "10",
19
          "111",
          "0110",
20
          "1101",
21
          "010",
22
          "000",
23
          "1",
24
          "001",
25
          "0001",
26
          "011",
27
          "1001",
28
          "1011",
29
          "1100"};
30
31
32
     int main()
33
34
       char str[1010];
35
       gets(str);
36
37
       int len = strlen(str);
       for (int i = 0; i < len; i++)</pre>
38
39
       {
          if (str[i] == ' ')
40
```

```
44 }
45 return 0;
47 }
```

F.我的名字

第二道签到题。

只要注意**不分大小写**即可。

```
1
    #include <stdio.h>
 2
    #include <string.h>
 3
4
     int main()
 5
     {
       char s[10010];
6
7
       scanf("%s", s);
8
9
       int len = strlen(s);
10
       int m = 0, t = 0, l = 0, f = 0; // 分别记录mtlf的个数
       for (int i = 0; i < len; i++)</pre>
11
12
         if (s[i] == 'M' || s[i] == 'm')
13
14
           m++;
15
         else if (s[i] == 'T' || s[i] == 't')
16
         else if (s[i] == 'L' || s[i] == 'l')
17
18
         else if (s[i] == 'F' || s[i] == 'f')
19
20
           f++;
       }
21
22
       while (m || t || l || f)
23
24
         if (m > 0)
25
26
           printf("m");
27
28
           m--;
29
         }
         if (t > 0)
30
31
32
           printf("t");
33
           t--;
34
         }
         if (1 > 0)
35
36
37
           printf("1");
38
           1--;
39
         }
         if (f > 0)
40
41
           printf("f");
42
43
```

```
47 return 0;
48 }
```

G.约好了下午三点,在记忆博物馆

首先非常抱歉题目测试数据出了问题TT,会导致某些算法WA,题解这里是直接模拟。 这里先分享2206的wmh同学的解法,我认为比我写的要更简单一点。 稍后会把wmh同学的文字题解发在评论区。

```
1
     #include<stdio.h>
 2
 3
     int main()
 4
         int n,T,t=0,num=1,i,station[13][2]={0},s=0;
 5
 6
         scanf("%d%d",&n,&T);
 7
         for(i=0;i<T;i++){</pre>
 8
              int a,b,x;
              scanf("%d%d%d",&a,&b,&x);
 9
              station[a][0]=b;
10
              station[a][1]=x;
11
12
         }
13
         while(1){
              S++;
14
              if(s>100){
15
16
                  num=0;
                  break;
17
18
              }
              t+=station[num][1];
19
              num=station[num][0];
20
              if(num==0||num==n)
21
                  break:
22
23
         }
         if(num!=n)
24
              printf("No\n0");
25
         else if(num==n&&t>120)
26
              printf("No\n%d",t);
27
         else
28
29
              printf("yes\n%d",t);
30
         return 0;
     }
31
```

有一点难度,但是不涉及算法(也可以使用最短路的方法,但是有点大材小用)。 这里巧妙定义一个二维数组flag[13][13],flag[a][b]即可表示为一条从a站到b站的列车的用时。 之后只需要模拟判断是否到站/无路可走即可(看代码)。

需要注意的是**每个站点只能作为一次起点**这个条件十分重要,因为这样你到达记忆博物馆的路线只能是一条或没有。

```
#include <stdio.h>
#include <string.h>

int main()
```

```
8
      scanf("%d%d", &n, &t);
9
10
      while (t--)
11
12
        int a, b, x;
        scanf("%d%d%d", &a, &b, &x);
13
14
        flag[a][b] = x;
      }
15
16
      int crt = 1, cnt = 0; // 分别为当前所在位置和用时
17
      while (1) // 一个死循环, break退出
18
19
      {
        int f = 0; // 判断是否找到路
20
        for (int i = 1; i <= 12; i++)
21
22
          if (flag[crt][i])
23
24
          {
25
             cnt += flag[crt][i];
             flag[crt][i] = 0; // 使用后归零,防止死循环
26
27
             crt = i;
             f = 1;
28
29
             break;
30
          }
31
         if (!f) // 没找到路直接输出No结束
32
33
          printf("No\n0");
34
35
          return 0;
36
         if (crt == n)
37
38
          if (cnt <= 120)
39
40
             printf("yes\n%d", cnt);
41
42
             return 0;
           }
43
          else
44
45
             printf("No\n%d", cnt);
46
47
             return 0;
48
          }
         }
49
50
       }
51
    }
```

H.唯一的道标

数学题,直接比较所在位置和四个出口的距离即可。 需要注意的是浮点数和输出格式是没有小数点及之后的数的。

```
#include <stdio.h>
#include <string.h>
#include <math.h>
```

```
return (a - c) * (a - c) + (b - d) * (b - d);
 7
8
     }
9
     int main()
10
11
     {
       double n, m, a, b;
12
13
       scanf("%lf%lf%lf%lf", &n, &m, &a, &b);
14
       double x[4] = \{n / 2, n, n / 2, 0\}, y[4] = \{0, m / 2, m, m / 2\};
15
       double min = caculateDistance(a, b, x[0], y[0]);
16
       int flag = 0;
17
       for (int i = 1; i < 4; i++)</pre>
18
19
         double temp = caculateDistance(a, b, x[i], y[i]);
20
         if (min > temp)
21
22
23
           min = temp;
           flag = i;
24
25
         }
       }
26
27
28
       printf("(%.0f, %.0f)", x[flag], y[flag]);
29
30
       return 0;
31
     }
                                                                                            举报
```

C) (C++





○↑ 默认排序 ~ 0条评论



没有回复

请留下你的观点吧~

发布

/ 牛客博客,记录你的成长

关于博客 意见反馈 免责声明 牛客网首页