

## 附件3：U18 组样题及说明(C++)

比赛按流程分为：初赛——网络赛、决赛——现场团队赛、现场个人赛；

U18 组是面向年龄在 2020 年 9 月 1 日前未满 19 周岁的青少年设置的比赛组别；

此组别的测试题说明与题目样例如下：

### 1 试题说明

1 试题 U18 - A 用于 U18 组的网络个人赛，试题按类型分为：选择题、填空题、程序完善题和程序设计题，其中程序设计题评测方式与 NOI 类似，对于每道题目，选手只能提交一次代码，以此代码作为统一评测的答卷，测试时为每道题提供了 10 - 20 组测试数据，根据数据点判断得分。

2 试题 U18 - B 用于 U18 组的现场团队赛，考核题目为程序设计题，题目评测方式与 ICPC 类似，对于每道题目，每支队伍可以多次提交，采取实时反馈与罚时机制；

3 试题 U18 - C 用于 U18 组的现场个人赛，考核题目为程序设计题，评测方式与 IOI 类似，对于每道题目，每位选手可以多次提交，采取实时反馈与得分机制；

### 2 题目样例

#### U18 - A 选择题（样例）

##### 题目

在二进制下， $1011001 + (\quad) = 1100110$ 。

- A. 1011
- B. 1101
- C. 1010
- D. 1111

##### 解题思路

二进制正整数的加法问题，直接用减法计算即可。

##### 参考答案

1 | B

#### U18 - A 填空题（样例）

##### 题目

书架上有 10 本书，编号从 1 到 10，从其中选 3 本，其中每两本的编号都不相邻的选法一共有( )种。

##### 解题思路

设从 10 个本书中任意抽走 3 本，剩下 7 本，抽走的书，应该原本的位置，应该在这 7 本书之间的 8 个间隔才符合要求，即可以将模型转换为：将 3 本书放在 8 个位置中的组合数，所以答案是  $C_8^3$ 。

##### 参考答案

1 | 56

#### U18 - A 程序完善题（样例）

## 题目

有一个小国家，国家内有  $n$  座城市和  $m$  条双向的道路，每条道路连接着两座不同的城市。其中 1 号城市为国家的首都。由于地震频繁可能导致某一个城市与外界交通全部中断。这个国家的首脑想知道，如果只有第  $i$  ( $i > 1$ ) 个城市因地震而导致交通中断时，首都到多少个城市的最短路径长度会发生改变。如果因为无法通过第  $i$  个城市而导致从首都出发无法到达某个城市，也认为到达该城市的最短路径长度改变。对于每一个城市  $i$ ，假定只有第  $i$  个城市与外界交通中断，输出有多少个城市会因此导致到首都的最短路径长度改变。我们采用邻接表的方式存储图的信息，其中  $head[x]$  表示顶点  $x$  的第一条边的编号， $next[i]$  表示第  $i$  条边的下一条边的编号， $point[i]$  表示第  $i$  条边的终点， $weight[i]$  表示第  $i$  条边的长度。

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 #define MAXN 6000
5 #define MAXM 100000
6 #define infinity 2147483647
7 int head[MAXN], next[MAXM], point[MAXM], weight[MAXM];
8 int queue[MAXN], dist[MAXN], visit[MAXN];
9 int n, m, x, y, z, total = 0, answer;
10 void link(int x, int y, int z)
11 {
12     total++;
13     next[total] = head[x];
14     head[x] = total;
15     point[total] = y;
16     weight[total] = z;
17     total++;
18     next[total] = head[y];
19     head[y] = total;
20     point[total] = x;
21     weight[total] = z;
22 }
23 int main()
24 {
25     int i, j, s, t;
26     cin >> n >> m;
27     for (i = 1; i <= m; i++)
28     {
29         cin >> x >> y >> z;
30         link(x, y, z);
31     }
32     for (i = 1; i <= n; i++)
33         dist[i] = infinity;
34     (1)_____ ;
35     queue[1] = 1;
36     visit[1] = 1;
37     s = 1;
38     t = 1;
39     // 使用 SPFA 求出第一个点到其余各点的最短路长度
40     while (s <= t)
41     {
42         x = queue[s % MAXN];
43         j = head[x];
44         while (j != 0)
45         {
46             if ((2)_____)
```

```

47     {
48         dist[point[j]] = dist[x] + weight[j];
49         if (visit[point[j]] == 0)
50         {
51             t++;
52             queue[t % MAXN] = point[j];
53             visit[point[j]] = 1;
54         }
55     }
56     j = next[j];
57 }
58 (3)_____ ;
59 s++;
60 }
61 for (i = 2; i <= n; i++)
62 {
63     queue[1] = 1;
64     memset(visit, 0, sizeof(visit));
65     visit[1] = 1;
66     s = 1;
67     t = 1;
68     while (s <= t)
69     { // 判断最短路长度是否不变
70         x = queue[s];
71         j = head[x];
72         while (j != 0)
73         {
74             if (point[j] != i && (4)_____ && visit[point[j]] == 0)
75             {
76                 (5)_____ ;
77                 t++;
78                 queue[t] = point[j];
79             }
80             j = next[j];
81         }
82         s++;
83     }
84     answer = 0;
85     for (j = 1; j <= n; j++)
86         answer += 1 - visit[j];
87     cout << i << ":" << answer - 1 << endl;
88 }
89 return 0;
90 }

```

### 参考答案

- 1 (1) dist[1]=0
- 2 (2) dist[x]+weight[j]<dist[point[j]]
- 3 (3) visit[x]=0
- 4 (4) weight[j]+dist[x]==dist[point[j]]
- 5 (5) visit[point[j]]=1

## 仓储管理

### 题目描述

小鹏是一个能干的仓储管理员。他负责鹏城的防疫物资调动工作。在自动化考察团的检查中，小鹏展示了自主研发的统计系统。

物资账目按  $1, 2, 3, \dots$  的编号记录，考察团会提出若干个提问，问题是：在第  $a$  到  $b$  笔账中，最少的一笔进帐是多少。

在考察团提出问题的过程中，系统还会受到其他实时指令，不断更新数据。

请你尝试模拟出这个统计系统的基础功能。

### 输入格式

第一行有两个数  $m$  和  $n$ ，表示有  $m$  笔账和有  $n$  个操作；

第二行有  $m$  个正整数，表示  $m$  笔帐的记录情况；

接下来  $n$  行，每行三个整数，分别表示  $p, x, y$ ：

当  $p = 1$  时，表示这是一个来自考察团的提问，查询第  $x$  到第  $y$  笔账的信息；

当  $p = 2$  时，表示这是一个实时修改指令，将第  $x$  个数字修改成了  $y$ ；

### 输出格式

一行，包含若干个整数，任意两数中间用单个空格隔开，表示系统对每个提问对应的答案。

### 样例输入

```
1 | 10 3
2 | 1 2 3 4 5 6 7 8 9 10
3 | 1 2 7
4 | 2 2 5
5 | 1 2 10
```

### 样例输出

```
1 | 2 3
```

### 样例分析

总共有 10 笔帐目，3 个操作：

第一次操作是询问，在 2 - 7 笔帐中，最小值是 2；

第二次操作是修改，把第 2 笔账改成了 5；

第三次操作是询问，在 2 - 10 笔账中，最小值是 3；

### 数据范围

对于 30% 的数据满足： $1 \leq n, m \leq 1000$ 。

对于 100% 的数据满足： $1 \leq n, m \leq 100000$ 。

### 解题思路

题目是经典的 *RMQ* 问题，考核的是对一个数列进行改点求点的操作，因为数据范围比较大，所以可以使用线段树或者树状数组等数据结构来辅助完成。

### 参考代码

```

1 //线段树实现 RMQ
2 #include<bits/stdc++.h>
3 using namespace std;
4 const int N=1e5+5;
5 int n,m;
6 struct nod{
7     int l,r,val=0;
8 }a[N*4];
9
10 void bt(int x,int l,int r){
11     a[x].l=l;
12     a[x].r=r;
13     if(l==r) return ;
14
15     int mid=(l+r)>>1;
16     bt(x*2, l,mid);
17     bt(x*2+1,mid+1, r);
18 }
19
20 void chg(int x,int pos,int k){
21     if(a[x].l==a[x].r){
22         a[x].val=k;
23         return ;
24     }
25     //分治修改
26     int mid=(a[x].l+a[x].r)>>1;
27     if(pos<=mid) chg(x*2, pos,k);
28     else chg(x*2+1,pos,k);
29     //回溯更新
30     a[x].val=min(a[x*2].val,a[x*2+1].val);
31 }
32
33 int que(int x,int l,int r){
34     if(a[x].l==l&&a[x].r==r) return a[x].val;
35     int mid=(a[x].l+a[x].r)>>1;
36
37     if(r<=mid) return que(x*2,l,r);
38     else if(mid<l) return que(x*2+1,l,r);
39     else return min(que(x*2,l,mid),que(x*2+1,mid+1,r));
40 }
41
42 int main(){
43     cin>>n>>m;
44     bt(1,1,n);
45     for(int x,i=1;i<=n;i++){
46         scanf("%d",&x);
47         chg(1,i,x);
48     }
49     int k,x,y;
50     while(m--){
51         scanf("%d %d %d",&k,&x,&y);
52         if(k==1) printf("%d ",que(1,x,y));
53         if(k==2) chg(1,x,y);
54     }
55     return 0;
56 }

```

