

附件3：U18 组样题及说明(Python)

BBCC 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 import numpy
2 import math
3
4 MAXN = 6001
5 MAXM = 100001
6
7 n = int(input("n=?"))
8 m = int(input("m=?"))
9
10 head = [0] * MAXN
11 nextl = [0] * MAXM
12 point = [0] * MAXM
13 weight = [0] * MAXM
14 total = 0
15
16 def link(x, y, z):
17     global total
18     total += 1
19     nextl[total] = head[x]
20     head[x] = total
21     point[total] = y
22     weight[total] = z
23     total += 1
24     nextl[total] = head[y]
25     head[y] = total
26     point[total] = x
27     weight[total] = z
28     total += 1
29
30 for i in range(m):
31     x, y, z = list(map(int, input("x, y, z=(以逗号分隔)").split(',')))
32     link(x, y, z)
33
34 dist = [math.inf] * (n+1)
35 visit = np.zeros(n+1, dtype=int)
36 queue = [0] * MAXN
37
38 (1)_____
39 queue[1] = 1
40 visit[1] = 1
41 s = 1
42 t = 1
43
44 # 使用 SPFA 求出第一个点到其余各点的最短路长度
45 while (s <= t):
46     x = queue[s % MAXN]
```

```

47     j = head[x]
48     while (j != 0):
49         if (2)_____:
50             dist[point[j]] = dist[x] + weight[j]
51             if visit[point[j]] == 0:
52                 t += 1
53                 queue[t % MAXN] = point[j]
54                 visit[point[j]] = 1
55             j = nextl[j]
56     (3)_____
57     s += 1
58
59 for i in range(2, n+1):
60     queue[1] = 1
61     visit.fill(0)
62     visit[1] = 1
63     s = 1
64     t = 1
65     while (s <= t):
66         # 判断最短路径长度是否不变
67         x = queue[s]
68         j = head[x]
69         while (j != 0):
70             if (point[j] != i) and ((4)_____) and (visit[point[j]] ==
0):
71                 (5)_____
72                 t += 1
73                 queue[t] = point[j]
74                 j = nextl[j]
75             s += 1
76
77     answer = 0
78     for j in range(1, n+1):
79         answer += 1 - visit[j]
80     print(f"{i}:{answer-1}")

```

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

```

U18 程序设计题（样例）

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, ... 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 import numpy as np
2
3 N = int(1e5 + 5)
4 a = np.zeros((N * 4, 3), dtype=int)
5
6 def bt(x, l, r):
7     a[x][0] = l
8     a[x][1] = r
9     if (l == r):
10        return
11    mid = int((l + r) / 2)
12    bt(x * 2, l, mid)
13    bt(x * 2 + 1, mid + 1, r)
14
15 def chg(x, pos, k):
16    if (a[x][0] == a[x][1]):
17        a[x][2] = k
18        return
19    # 分治修改
20    mid = int((a[x][0] + a[x][1]) / 2)
21    if (pos <= mid):
22        chg(x * 2, pos, k)
23    else:
24        chg(x * 2 + 1, pos, k)
25    # 回溯更新
26    a[x][2] = min(a[x * 2][2], a[x * 2 + 1][2])
27
28 def que(x, l, r):
29    if (a[x][0] == l) and (a[x][1] == r):
30        return a[x][2]
31    mid = int((a[x][0] + a[x][1]) / 2)
32    if (r <= mid):
33        return que(x*2, l, r)
34    elif (mid < l):
35        return que(x*2+1, l, r)
36    else:
37        return min(que(x*2, l, mid), que(x*2+1, mid+1, r))
38
39 n = int(input("n=?"))
40 m = int(input("m=?"))
41 bt(1, 1, n)
42
43 x = list(map(int, input("n笔账=? (以逗号分隔)").split(',')))
44 for i in range(1, n+1):
45    chg(1, i, x[i-1]);
46
47 ans = []
48 while (m > 0):
49    m -= 1
50    k, x, y = list(map(int, input("k, x, y=? (以逗号分隔)").split(',')))
51    if (k == 1):
52        ans.append(que(1, x, y))
53    if (k == 2):
54        chg(1, x, y)
55
56 print(" ".join(list(map(str, ans))))

```

