

第一堂課 小測驗題解

日月卦長



連結

1. <https://codeforces.com/contestInvitation/3ac99153b0297aafe01c6709513c80bee33cdef0>
2. [Dashboard - The qualification test for 2024 NTHU CP II - Codeforces](#)

A. All Pairs Bottleneck

- 給一張無向完全圖 G ，其中 $G[a][b]$ 表示 a 到 b 的邊權 (正整數)
- 定義一條路徑 $P = \{(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)\}$ 的瓶頸
 $bottleneck(P) = \min\{G[v_1][v_2], G[v_2][v_3], \dots, G[v_{n-1}][v_n]\}$
- 定義 a 到 b 的最大瓶頸路徑：
$$Bottleneck(a, b) = \max_{P: a \rightarrow b} \{bottleneck(P)\}$$
- 請你輸出所有點對的 $Bottleneck$

作法一：Floyd 最短路徑稍微修改

```
for(int k = 0; k < n; ++k)
    for(int i = 0; i < n; ++i)
        for(int j = 0; j < n; ++j)
            G[i][j] = max(G[i][j], min(G[i][k], G[k][j]));
```

作法二：最大瓶頸樹 = 最大生成樹

1. 建立最大生成樹 T
2. 由於樹上任兩點之間的路徑就是最大瓶頸路徑
直接計算路徑上最小權重的邊就可以了

B,C. Merry Go Round

- 給一張有向圖
- 問你從編號 1 的點開始經過其他所有點各一次最後回到編號 1 的點有幾種不同路徑
- 注意這題有重邊
- Easy: $n \leq 7$
- Hard: $n \leq 20$

解法一：暴力枚舉 (只能通過Easy version)

- 設 $\text{adj}[a][b]$ 表示 a 到 b 有幾條邊相連

```
vector<int> perm(n - 1);  
// fill with 2, ..., n  
iota(perm.begin(), perm.end(), 2);  
int ans = 0;  
do {  
    int path_num = 1;  
    path_num *= adj[1][perm.front()];  
    for (size_t i = 1; i < perm.size(); ++i)  
        path_num *= adj[perm[i - 1]][perm[i]];  
    path_num *= adj[perm.back()][1];  
    ans += path_num;  
} while (next_permutation(perm.begin(), perm.end()));  
cout << ans << '\n';
```

解法二：位元動態規劃

- 這是非常經典的旅行銷售員問題
- 各位網路上隨便查都可以找到答案
- 或是可以參考上學期競程一的內容

D. Number of Minimum Number

- 給一個陣列 a_1, a_2, \dots, a_n ，以及 q 個操作
- 操作有兩種
 1. 給定 k, v ，將 a_k 更新成 $a_k + v$
 2. 給定 L, R ，計算 a_L, a_{L+1}, \dots, a_R 中的最小值以及最小值出現了幾次

解法：線段樹或分塊

```
struct Node {  
    int Min, Num;  
    Node(int Min, int Num = 1) : Min(Min), Num(Num) {}  
    Node operator+(const Node& other) const {  
        if (Min < other.Min) {  
            return *this;  
        }  
        if (Min > other.Min) {  
            return other;  
        }  
        return Node(Min, Num + other.Num);  
    }  
};
```