



逃走経路 (Escape Route)

IOI 国の 1 日は S 個の単位時間に分割されており、単位としてビヨウを用いている。1 日の始まりから x ビヨウ ($0 \leq x < S$) 経過した時点時刻 x とする。IOI 国は N 個の都市からなり、 0 から $N-1$ までの番号が付けられている。また、都市と都市を結ぶ道は M 本あり、 0 から $M-1$ までの番号が付けられている。どの都市からどの都市へも何本かの道を通ることによって移動することができる。道 i ($0 \leq i \leq M-1$) は都市 A_i と都市 B_i を双方向に結んでおり、ちょうど L_i ビヨウかけて通行することができる。また、毎日時刻 C_i からその日が終わるまで道 i 全体で一斉検問が行われている。

秘密結社 JOI 団は IOI 国で暗躍する組織の 1 つである。秘密結社という組織の性質上、構成員は表の人間に知られるわけにはいかない。これは、IOI 国において毎日行われている一斉検問に、構成員が遭遇してはならないということを意味する。したがって、JOI 団の構成員が道 i を通行するには、 $0 \leq x \leq C_i - L_i$ を満たす時刻 x に都市 A_i と都市 B_i のいずれかから出発し、時刻 $x + L_i$ にもう一方の都市に到着する必要がある。検問は都市では行われないので、道 i の一斉検問が行われている最中に、都市 A_i や都市 B_i に構成員がいてもよい。

JOI 団の構成員は Q 人おり、 0 から $Q-1$ までの番号が付けられている。構成員 j ($0 \leq j \leq Q-1$) はある日の時刻 T_j に都市 U_j を出発し、都市 V_j を目指して移動を開始する。構成員は移動の最中に都市で待機することもできる。都市 V_j にたどり着くまでに複数日かかるかもしれない。

IOI 国の都市と道の情報、一斉検問の情報、JOI 団の構成員の情報が与えられたとき、それぞれの j ($0 \leq j \leq Q-1$) について、構成員 j が都市 U_j を出発してから都市 V_j にたどり着くまでにかかる時間の最小値を求めよ。

実装の詳細

この課題の採点は、入出力の高速化のため採点プログラム `grader` を用いて行う。

あなたは 1 つのファイルを提出しなければならない。ファイルは `escape_route.cpp` という名前である。このファイルは以下の関数を実装していなければならない。また、`#include` プリプロセッサ指令によって `escape_route.h` を読み込むこと。

- `std::vector<long long> calculate_necessary_time(`
 `int N, int M, long long S, int Q, std::vector<int> A, std::vector<int> B,`
 `std::vector<long long> L, std::vector<long long> C, std::vector<int> U,`
 `std::vector<int> V, std::vector<long long> T)`

この関数は、各テストケースにつき 1 回だけ呼び出される。

- 引数 N は、IOI 国の都市の個数である。



- 引数 M は、IOI 国の道の本数である。
- 引数 S は、IOI 国の 1 日が S ビヨウであることを表す。
- 引数 Q は、JOI 団の構成員の人数である。
- 引数 A, B, L, C は長さ M の配列であり、道 i は都市 $A[i]$ と都市 $B[i]$ を結び、ちょうど $L[i]$ ビヨウかけて通行することができ、時刻 $C[i]$ から一斉検問が始まることを表す ($0 \leq i \leq M-1$)。
- 引数 U, V, T は長さ Q の配列であり、構成員 j は時刻 $T[j]$ に都市 $U[j]$ を出発し、都市 $V[j]$ を目指して移動を開始することを表す ($0 \leq j \leq Q-1$)。
- この関数は長さ Q の long long 型の配列 `answer` を返さなければならない。すべての $0 \leq j \leq Q-1$ に対して、構成員 j が都市 $U[j]$ を出発してから都市 $V[j]$ にたどり着くまでにかかる時間の最小値は `answer[j]` ビヨウであることを表す。

重要な注意

- あなたの提出したプログラムは、標準入力・標準出力、あるいは他のファイルといかなる方法でもやりとりしてはならない。ただし、標準エラー出力にデバッグ情報等を出力することは許される。
- 採点プログラムは入力の形式のチェックを行わない。形式の異なる入力を与えた場合の動作は保証されない。
- `calculate_necessary_time` が返す配列の長さが Q でなかったり、配列の要素に負の数が含まれている場合は、採点プログラムの動作は保証されない。

コンパイル・実行の方法

作成したプログラムをテストするための、採点プログラムが、コンテストサイトからダウンロードできるアーカイブの中に含まれている。このアーカイブには、提出しなければならないファイルのサンプルも含まれている。

採点プログラムは 1 つのファイルからなる。そのファイルは `grader.cpp` である。作成したプログラムをテストするには、`grader.cpp`, `escape_route.cpp`, `escape_route.h` を同じディレクトリに置き、次のようにコマンドを実行する。

```
g++ -std=gnu++17 -O2 -fsigned-char -o grader grader.cpp escape_route.cpp
```

コンパイルが成功すれば、`grader` という実行ファイルが生成される。

この課題においては、配布される採点プログラムと CMS での評価に使われる採点プログラムは同一である。



採点プログラムの入力

採点プログラムは標準入力から以下の形式で入力を読み込む。入力はすべて整数でなければならない。

```
N M S Q
A0 B0 L0 C0
⋮
AM-1 BM-1 LM-1 CM-1
U0 V0 T0
⋮
UQ-1 VQ-1 TQ-1
```

採点プログラムの出力

採点プログラムは標準出力に Q 行出力をする。 $k+1$ 行目 ($0 \leq k \leq Q-1$) には、 `answer[k]` を出力する。

制約

- $2 \leq N \leq 90$.
- $N-1 \leq M \leq \frac{N(N-1)}{2}$.
- $2 \leq S \leq 1\,000\,000\,000\,000\,000 = 10^{15}$.
- $1 \leq Q \leq 3\,000\,000$.
- $0 \leq A_i \leq N-1$ ($0 \leq i \leq M-1$).
- $0 \leq B_i \leq N-1$ ($0 \leq i \leq M-1$).
- $A_i \neq B_i$ ($0 \leq i \leq M-1$).
- $(A_i, B_i) \neq (A_k, B_k)$, $(A_i, B_i) \neq (B_k, A_k)$ ($0 \leq i < k \leq M-1$).
- $1 \leq L_i < S$ ($0 \leq i \leq M-1$).
- $L_i \leq C_i < S$ ($0 \leq i \leq M-1$).
- どの2つの都市の間も、いくつかの道を通ることで行き来可能である。
- $0 \leq U_j \leq N-1$ ($0 \leq j \leq Q-1$).
- $0 \leq V_j \leq N-1$ ($0 \leq j \leq Q-1$).
- $U_j \neq V_j$ ($0 \leq j \leq Q-1$).
- $0 \leq T_j < S$ ($0 \leq j \leq Q-1$).



小課題

1. (5 点) $N \leq 40$, $Q \leq 1000$.
2. (20 点) $N \leq 40$, $U_j = 0$ ($0 \leq j \leq Q - 1$).
3. (10 点) $N \leq 40$.
4. (35 点) $N \leq 60$.
5. (30 点) 追加の制約はない.

入出力例

入力例 1	出力例 1
4 5 20 6	3
0 1 3 19	8
0 2 2 8	14
1 2 4 15	2
1 3 5 14	5
2 3 1 18	7
0 3 5	
0 3 7	
0 3 9	
2 0 6	
3 1 10	
1 2 15	

この入力例は小課題 1, 3, 4, 5 の制約を満たす.

構成員 0 は時刻 5 に都市 0 を出発し, 都市 3 に移動する. 以下のように移動すると 3 ビョウかかる.

- 道 1 を使用し, 時刻 5 に都市 0 を出発して, 時刻 7 に都市 2 に到着する.
- 道 4 を使用し, 時刻 7 に都市 2 を出発して, 時刻 8 に都市 3 に到着する.

これが最小値なので, $\text{answer}[0] = 3$ である.

構成員 1 は時刻 7 に都市 0 を出発し, 都市 3 に移動する. 以下のように移動すると 8 ビョウかかる.

- 道 0 を使用し, 時刻 7 に都市 0 を出発して, 時刻 10 に都市 1 に到着する.
- 道 2 を使用し, 時刻 10 に都市 1 を出発して, 時刻 14 に都市 2 に到着する.



- 道 4 を使用し、時刻 14 に都市 2 を出発して、時刻 15 に都市 3 に到着する。

これが最小値なので、 $\text{answer}[1] = 8$ である。

構成員 2 は時刻 9 に都市 0 を出発し、都市 3 に移動する。以下のように移動すると、翌日の時刻 3 に都市 3 に到着する。合計 14 ビョウかかる。

- 翌日の時刻 0 まで都市 0 で待機する。
- 道 1 を使用し、時刻 0 に都市 0 を出発して、時刻 2 に都市 2 に到着する。
- 道 4 を使用し、時刻 2 に都市 2 を出発して、時刻 3 に都市 3 に到着する。

これが最小値なので、 $\text{answer}[2] = 14$ である。

入力例 2	出力例 2
6 10 100 9	42
5 3 4 29	32
1 0 6 26	4
0 4 2 7	93
0 5 18 18	99
2 0 79 82	6
3 4 35 46	102
1 2 15 57	60
2 4 3 6	39
4 1 21 83	
3 2 47 53	
0 2 63	
0 4 70	
0 4 98	
0 5 25	
0 5 19	
0 4 96	
0 5 2	
0 3 62	
0 3 83	

この入力例はすべての小課題の制約を満たす。



The 20th Japanese Olympiad in Informatics (JOI 2020/2021)
Spring Training Camp/Qualifying Trial
March 20–23, 2021 (Komaba, Tokyo)

Contest Day 2 – Escape Route

入力例 3	出力例 3
8 12 100000000000000000 13	72937946261976
2 0 4451698272827 120985696255786	929038398222642
6 5 78520421713825 342652131468508	702857945988825
2 1 185377268405175 382583457603811	272921388674172
0 4 54350742205838 133614919589507	580895059624855
7 0 68486247989149 651590905094148	181808439529442
0 6 85177550834829 299184420663240	117602869946965
5 2 442329739732459 926608308293721	569788353034530
3 7 78020232822359 913548478810253	1181546234307589
1 3 267796317244889 687571310475622	244230056736534
5 4 90590208828121 910324397566584	513790925121797
5 7 8414633059584 17796117322043	617759130113052
4 6 45682367792138 204548471584556	674500988551485
7 2 44779065000162	
3 5 79376234836942	
4 7 305556687070759	
4 3 927935834343174	
5 1 663284649258985	
2 5 967584209777344	
5 2 963749709374595	
7 4 484562389171308	
1 5 446160773830045	
6 4 801452311055604	
3 1 744524289545354	
0 6 467418420721777	
5 6 371181379240653	

この入力例は小課題 1, 3, 4, 5 の制約を満たす。