



## 長距離バス (Long Distance Coach)

IOI国には都市Iと都市Oを結ぶ長距離バスが走っている。バスの車内には給水器が設置されており、乗客や運転手は水を飲むことができる。バスは時刻0に都市Iを出発し、時刻 $X$ に都市Oに到着する。途中には、水を補給できる地点が $N$ 個ある。バスが $i$ 番目 ( $1 \leq i \leq N$ ) の補給地点を訪れる時刻は $S_i$ である。

最初、給水器には水は入っていない。バスの出発前に水を補給することができる。また、補給地点においても水を補給することができる。水を補給するコストは、場所によらず、1リットルあたり $W$ 円である。

このバスには都市Iで $M$ 人の乗客が乗車し、乗客には1から $M$ の番号が付けられている。都市I以外の場所では、新たな乗客が乗車することはない。乗客 $j$  ( $1 \leq j \leq M$ ) は時刻 $D_j$ に1リットルの水を必要とする。また、水を必要としてから時間 $T$ が経過する度に、再び1リットルの水を必要とする。すなわち、乗客 $j$ が水を必要とするのは、時刻 $D_j + kT$  ( $k = 0, 1, 2, \dots$ ) においてである。ここで、 $1 \leq D_j < T$ であり、 $T$ の値は乗客によらず一定である。乗客が水を必要としたときに給水器に水が残っていないと、その乗客は途中下車してしまう。乗客 $j$ が都市Oに到着する前に途中下車した場合、運賃の払い戻しとして $C_j$ 円のコストがかかる。

運転手も水を必要とする。運転手は時刻0に1リットルの水を必要とし、乗客と同様、水を必要としてから時間 $T$ が経過する度に、再び1リットルの水を必要とする。すなわち、運転手が水を必要とするのは、時刻 $kT$  ( $k = 0, 1, 2, \dots$ ) においてである。運転手が水を必要としたときに給水器に水が残っていないと、バスの運行を続けることができない。

2人以上の人が同時に水を必要とすることはない。また、バスが都市Oや補給地点を訪れると同時に、乗客や運転手が水を必要とすることはない。

補給地点における水の補給量を調整することで、水の補給にかかるコストと、運賃の払い戻しにかかるコストの合計を小さくして、バスを都市Oまで運行したい。あなたの仕事は、バスの運行中に、どの地点でどれだけ水を補給するかを決定することである。

## 課題

バスの運行にかかる時間、補給地点の情報、乗客および運転手の情報が与えられたとき、バスを都市Oまで運行するための、水の補給にかかるコストと、運賃の払い戻しにかかるコストの合計の最小値を求めるプログラムを作成せよ。

## 入力

標準入力から以下のデータを読み込め。

- 1行目には、5個の整数 $X, N, M, W, T$ が空白を区切りとして書かれている。これは、バスが時刻 $X$ に都市Oに到着し、補給地点が $N$ 個あり、バスに乗客が $M$ 人いて、水1リットルの補給にかかるコストが $W$ 円であって、乗客や運転手が水を必要とする間隔が $T$ であることを表す。



- 続く  $N$  行のうちの  $i$  行目 ( $1 \leq i \leq N$ ) には、整数  $S_i$  が書かれている。これは、バスが  $i$  番目の補給地点を訪れるのが時刻  $S_i$  であることを表す。
- 続く  $M$  行のうちの  $j$  行目 ( $1 \leq j \leq M$ ) には、2 個の整数  $D_j, C_j$  が空白を区切りとして書かれている。これは、乗客  $j$  が最初に水を必要とする時刻が  $D_j$  であることと、乗客  $j$  が途中下車した場合のコストが  $C_j$  円であることを表す。

## 出力

標準出力に、必要なコストの合計の最小値を表す整数を 1 行で出力せよ。

## 制限

すべての入力データは以下の条件を満たす。

- $1 \leq X \leq 1\,000\,000\,000\,000$ .
- $1 \leq N \leq 200\,000$ .
- $1 \leq M \leq 200\,000$ .
- $1 \leq W \leq 1\,000\,000$ .
- $1 \leq T \leq X$ .
- $1 \leq S_i < X$  ( $1 \leq i \leq N$ ).
- $1 \leq D_j < T$  ( $1 \leq j \leq M$ ).
- $1 \leq C_j \leq 1\,000\,000\,000$  ( $1 \leq j \leq M$ ).
- $D_j$  ( $1 \leq j \leq M$ ) は互いに異なる。
- バスが都市  $O$  や補給地点を訪れると同時に乗客や運転手が水を必要とすることはない。

## 小課題

この課題では小課題は全部で 4 個ある。各小課題の配点および追加の制限は以下の通りである。

### 小課題 1 [16 点]

- $N \leq 8$ .
- $M \leq 8$ .



小課題 2 [30 点]

- $N \leq 100$ .
- $M \leq 100$ .

小課題 3 [25 点]

- $N \leq 2000$ .
- $M \leq 2000$ .

小課題 4 [29 点]

追加の制限はない。

入出力例

入力例 1	出力例 1
19 1 4 8 7 10 1 20 2 10 4 5 6 5	103

この入力例において、出発前に 7 リットル、1 番目の補給地点で 4 リットルの水を補給する場合、バスの運行は次のように行われる。

1. バスが都市 I を出発する。この時点で給水器には 7 リットルの水が入っている。
2. 運転手および乗客 1, 2, 3, 4 がそれぞれ時刻 0, 1, 2, 4, 6 に水を 1 リットルずつ飲み、残りの水量は 2 リットルになる。
3. 運転手および乗客 1 がそれぞれ時刻 7, 8 に再び水を 1 リットルずつ飲み、残りの水量は 0 リットルになる。
4. 時刻 9 に乗客 2 が水を必要とするが、給水器に水が無いので途中下車する。
5. 時刻 10 に 1 番目の補給地点で水を 4 リットル補給する。残りの水量は 4 リットルになる。
6. 乗客 3, 4, 運転手, および乗客 1 がそれぞれ時刻 11, 13, 14, 15 に水を 1 リットルずつ飲み、残りの水量は 0 リットルになる。
7. 時刻 18 に乗客 3 が水を必要とするが、給水器に水が無いので途中下車する。



8. 時刻 19 にバスが都市 O に到着する.

水の補給は計 11 リットル行われるため給水にかかるコストは 88 円である. 乗客 2,3 が途中下車するコストは合計で 15 円である. これらの合計は 103 円である

102 円以下の合計コストでバスを最後まで運行することはできないので, 103 を出力する.

入力例 2	出力例 2
105 3 5 9 10	547
59	
68	
71	
4 71	
6 32	
7 29	
3 62	
2 35	

入力例 3	出力例 3
10000000000000 1 1 1000000 6	333333209997456789
999999259244	
1 123456789	



## 細長い屋敷 (Long Mansion)

JOI くんの家近くには細長い屋敷がある。この屋敷は部屋が東西に  $N$  個並んだ構造をしており、東から  $i$  番目の部屋を部屋  $i$  と呼ぶ。部屋  $i$  と部屋  $i+1$  ( $1 \leq i \leq N-1$ ) は廊下でつながっており、廊下は双方向に通行可能である。部屋と廊下を行き来するためには鍵が必要である。鍵には種類と呼ばれる番号が付けられている。同じ種類の鍵が複数あることもある。

部屋  $i$  や部屋  $i+1$  から、部屋  $i$  と部屋  $i+1$  を結ぶ廊下に入るには、種類  $C_i$  の鍵が必要である。

部屋  $i$  には  $B_i$  個の鍵が落ちている。それらの鍵の種類は  $A_{i,j}$  ( $1 \leq j \leq B_i$ ) である。JOI くんは、部屋に入るとその部屋に落ちている鍵を全て拾い、以後廊下に入るのに使うことができる。

鍵は何度でも使うことができる。また、同じ鍵が複数手に入ることがあるが、1つ持っているときと比べて特別役に立つことはない。

JOI くんは、この屋敷で迷子になったときに備えて、以下の形式のクエリに答えるプログラムを書くことにした。

- JOI くんが鍵を1つも持っていない状態で部屋  $x$  に迷い込んだとき、部屋  $y$  に到達することはできるか?

あなたの仕事は、JOI くん代わりにこのクエリに答えるプログラムを書くことである。

## 課題

屋敷の情報とクエリが与えられたとき、それぞれのクエリに対し、鍵を1つも持っていない状態で部屋に迷い込んだとき、部屋から部屋に到達することができるかを判定するプログラムを作成せよ。

## 入力

標準入力から以下の入力を読み込め。

- 1行目には整数  $N$  が書かれている。これは屋敷には部屋が  $N$  個あることを表す。
- 2行目には  $N-1$  個の整数  $C_1, C_2, \dots, C_{N-1}$  が空白を区切りとして書かれている。これらは、部屋  $i$  や部屋  $i+1$  から、部屋  $i$  と部屋  $i+1$  を結ぶ廊下に入るには種類  $C_i$  の鍵が必要であることを表す。
- 続く  $N$  行のうちの  $i$  行目 ( $1 \leq i \leq N$ ) には、1以上の整数  $B_i$  と、 $B_i$  個の整数  $A_{i,1}, A_{i,2}, \dots, A_{i,B_i}$  が空白を区切りとして書かれている。これは部屋  $i$  には鍵が  $B_i$  個落ちており、それらの鍵の種類が  $A_{i,j}$  ( $1 \leq j \leq B_i$ ) であることを表す。
- 次の行には整数  $Q$  が書かれている。これはクエリが  $Q$  個与えられることを表す。
- 続く  $Q$  行のうちの  $k$  行目 ( $1 \leq k \leq Q$ ) には、2個の整数  $X_k$  と  $Y_k$  が空白を区切りとして書かれている。これらは、 $k$  番目のクエリが、鍵を1つも持っていない状態で部屋  $X_k$  に迷い込んだとき、部屋  $X_k$  から部屋  $Y_k$  に到達できるかを尋ねるものであることを表す。



## 出力

標準出力に  $Q$  行で出力せよ。  $Q$  行のうちの  $k$  行目 ( $1 \leq k \leq Q$ ) には、鍵を 1 つも持っていない状態で部屋  $X_k$  に迷い込んだとき、部屋  $X_k$  から部屋  $Y_k$  に到達することができるなら YES を、そうでないなら NO を出力せよ。

## 制限

すべての入力データは以下の条件を満たす。

- $2 \leq N \leq 500\,000$ .
- $1 \leq Q \leq 500\,000$ .
- $1 \leq B_1 + B_2 + \dots + B_N \leq 500\,000$ .
- $1 \leq B_i \leq N$  ( $1 \leq i \leq N$ ).
- $1 \leq C_i \leq N$  ( $1 \leq i \leq N - 1$ ).
- $1 \leq A_{i,j} \leq N$  ( $1 \leq i \leq N$ ,  $1 \leq j \leq B_i$ ).
- $B_i$  個の整数  $A_{i,1}, \dots, A_{i,B_i}$  は互いに異なる ( $1 \leq i \leq N$ ).
- $1 \leq X_k \leq N$  ( $1 \leq k \leq Q$ ).
- $1 \leq Y_k \leq N$  ( $1 \leq k \leq Q$ ).
- $X_k \neq Y_k$  ( $1 \leq k \leq Q$ ).

## 小課題

この課題では小課題は全部で 4 個ある。各小課題の配点および追加の制限は以下の通りである。

### 小課題 1 [5 点]

- $N \leq 5\,000$ .
- $Q \leq 5\,000$ .
- $B_1 + B_2 + \dots + B_N \leq 5\,000$ .

### 小課題 2 [5 点]

- $N \leq 5\,000$ .
- $B_1 + B_2 + \dots + B_N \leq 5\,000$ .



### 小課題 3 [15 点]

- $N \leq 100\,000$ .
- $C_i \leq 20$  ( $1 \leq i \leq N - 1$ ).
- $A_{i,j} \leq 20$  ( $1 \leq i \leq N$ ,  $1 \leq j \leq B_i$ ).

### 小課題 4 [75 点]

追加の制限はない。

### 入出力例

入力例 1	出力例 1
5	YES
1 2 3 4	NO
2 2 3	NO
1 1	YES
1 1	
1 3	
1 4	
4	
2 4	
4 2	
1 5	
5 3	

- 1つ目のクエリでは、部屋 2, 1, 2, 3, 4 の順に移動することで、部屋 4 に到達することができる。
- 2つ目のクエリでは、部屋 3, 4 にしか行くことができず、手に入れられる鍵の種類は 1, 3 のみなので、部屋 2 に到達することができない。
- 3つ目のクエリでは、部屋 4 から部屋 5 に行く種類 4 の鍵を手に入れることができないため、部屋 5 に到達することができない。
- 4つ目のクエリでは、部屋 5, 4, 3 の順に移動することで、部屋 3 に到達することができる。



入力例 2	出力例 2
5	NO
2 3 1 3	YES
1 3	NO
1 2	YES
1 1	
1 3	
1 2	
4	
1 3	
3 1	
4 3	
2 5	

入力例 3	出力例 3
7	YES
6 3 4 1 2 5	NO
1 1	YES
1 5	
1 1	
1 1	
2 2 3	
1 4	
1 6	
3	
4 1	
5 3	
4 7	



## 自然公園 (Natural Park)

JOI 島は観光地であり、島全体が自然公園として指定されている。

JOI 島には  $N$  個の広場といくつかの道がある。広場には  $0$  から  $N-1$  までの番号が付けられている。道は島内の異なる 2 つの広場を結んでおり、双方向に移動可能である。どの広場についても、その広場と他の広場を結ぶ道は高々 7 本である。どの異なる 2 つの広場についても、それらを結ぶ道は高々 1 本しかない。また、どの 2 つの広場の間も、何本かの道を使うことで互いに移動できることが分かっている。

あなたは友達 IOI ちゃんと JOI 島を調査することになった。調査を効率的に行うために、あなたは島の構造を把握しなければならない。島内は野生動物が多く危険なので、運動能力の高い IOI ちゃんが島を探索し、IOI ちゃんの報告に基づいて、あなたが島の構造を特定することになった。

あなたは IOI ちゃんに対し、2 つの広場  $A, B$  と、経路可能な広場をいくつか指定し、広場  $A$  から広場  $B$  まで指定された広場のみを経由して移動できるか、という質問をする。IOI ちゃんは質問内容に従って島内を探索し、結果を報告する。

調査に長く時間をかけることはできないので、質問回数を 45 000 回以内に抑えたい。

## 課題

IOI ちゃんとやりとりをして、JOI 島の構造を特定するプログラムを作成せよ。

## 実装の詳細

あなたは、島の構造を特定する方法を実装した 1 個のプログラムを書かなければならない。プログラムは `park.h` をインクルードすること。

プログラムは、以下のルーチンを実装しなければならない。

- `void Detect(int T, int N)`

この関数は 1 回だけ呼び出される。

- 引数  $T$  は小課題番号を、 $N$  は広場の個数を表す。

また、プログラムの中では、以下の関数を呼び出して、特定した JOI 島の構造を出力しなければならない。

- `void Answer(int A, int B)`

この関数は、JOI 島の道の本数と等しい回数だけ呼び出されなければならない。

- 引数  $A, B$  は、広場  $A$  と広場  $B$  の間を結ぶ道があると特定したことを表す。



引数は以下の条件を満たさなければならない。

- $A, B$  は、 $0 \leq A < B \leq N - 1$  を満たさなければならない。これを満たさない場合、不正解 [1] となる。
- ある  $(A, B)$  の組を引数として呼び出した際、広場  $A$  と広場  $B$  の間には道がなければならない。これを満たさない場合、不正解 [2] となる。
- 同じ  $(A, B)$  の組を引数として 2 回以上呼び出してはならない。これを満たさない場合、不正解 [3] となる。

これに加え、プログラム中では以下の関数を呼び出すことができる。

• `int Ask(int A, int B, int Place[])`

この関数は、IOI ちゃんに質問をするために呼び出す関数である。

- `Place` は経路可能な広場を表す配列のポインタである。各  $i$  ( $0 \leq i \leq N - 1$ ) に対し、広場  $i$  が経路可能であるとき `Place[i] = 1` であり、広場  $i$  が経路不可能であるとき `Place[i] = 0` である。
- この関数の戻り値は、広場  $A$  から広場  $B$  まで配列 `Place[]` によって指定された広場のみを経由して移動できるとき 1 である。移動できないとき 0 である。

引数は以下の条件を満たさなければならない。

- $0 \leq A < B \leq N - 1$ .
- $0 \leq \text{Place}[i] \leq 1$  ( $0 \leq i \leq N - 1$ ).
- `Place[A] = 1`.
- `Place[B] = 1`.

これらの条件を満たさない場合、不正解 [4] となる。ただし、配列 `Place[]` の長さが  $N$  でないときの動作は保証されない。

また、関数 `Ask` を呼び出す回数が 45 000 回を超えてはならない。超えた場合、不正解 [5] となる。

関数 `Detect` の終了時点で、関数 `Answer` の引数として呼び出していない道が存在する場合、不正解 [6] となる。

内部での使用のために他の関数を実装したり、グローバル変数を宣言するのは自由である。ただし、あなたの提出は標準入力・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。

## 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、採点プログラムのサンプルが、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。



採点プログラムのサンプルは 1 つのファイルからなる。そのファイルは `grader.c` または `grader.cpp` である。作成したプログラムを `park.c` または `park.cpp` とするとき、作成したプログラムをテストするには、次のようにコマンドを実行する。

- C の場合

```
gcc -std=c11 -O2 -o grader grader.c park.c -lm
```

- C++ の場合

```
g++ -std=c++14 -O2 -o grader grader.cpp park.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。

実際の採点プログラムは、採点プログラムのサンプルとは異なることに注意すること。採点プログラムのサンプルは単一のプロセスとして起動する。このプログラムは、標準入力から入力を読み込み、標準出力に結果を出力する。

### 採点プログラムのサンプルの入力

採点プログラムのサンプルは標準入力から以下のデータを読み込む。

- 1 行目には整数  $T$  が書かれている。これは小課題番号を表す。
- 2 行目には整数  $N$  が書かれている。これは広場が  $N$  個あることを表す。
- 3 行目には整数  $M$  が書かれている。これは道が  $M$  本あることを表す。
- 続く  $M$  行のうちの  $i$  行目 ( $1 \leq i \leq M$ ) には、整数  $A_i, B_i$  が空白を区切りとして書かれている。これは広場  $A_i$  と広場  $B_i$  を双方向に結ぶ道があることを表す。

### 採点プログラムのサンプルの出力

採点プログラムのサンプルは標準出力へ以下の情報を出力する（引用符は実際には出力されない）。

- 正解と判定された場合、“Accepted” と出力される。
- プログラムの実行中に不正解と判定された場合、不正解の種類が “Wrong Answer [1]” のように出力され、実行が終了される。

実行するプログラムが複数の不正解の条件を満たした場合、表示される不正解の種類はそれらのうち 1 つのみである。



## 制限

すべての入力データは以下の条件を満たす。  $T, N, M$  の意味については、「採点プログラムのサンプルの入力」の項目を参照せよ。

- $1 \leq T \leq 5$ .
- $2 \leq N \leq 1400$ .
- $1 \leq M \leq 1500$ .
- どの広場についても、その広場と他の広場を結ぶ道は高々 7 本である。
- どの異なる 2 つの広場の間も、何本かの道を使うことで互いに移動できる。
- どの異なる 2 つの広場の間も、それらを結ぶ道は高々 1 本である。



## 小課題

この課題では小課題は全部で 5 個ある。各小課題の配点および追加の制限は以下の通りである。

### 小課題 1 [10 点]

- $T = 1$ .
- $N \leq 250$ .

### 小課題 2 [10 点]

- $T = 2$ .
- $M = N - 1$ .
- 広場 0 および広場  $N - 1$  について、その広場と他の広場を結ぶ道はちょうど 1 本ある。それ以外の広場については、その広場と他の広場を結ぶ道はちょうど 2 本ある。

### 小課題 3 [27 点]

- $T = 3$ .
- $M = N - 1$ .
- どの  $i (1 \leq i \leq N - 1)$  についても、広場 0 から広場  $i$  まで他に高々 8 個の広場のみを經由して移動可能である。

### 小課題 4 [30 点]

- $T = 4$ .
- $M = N - 1$ .

### 小課題 5 [23 点]

- $T = 5$ .



## やり取りの例

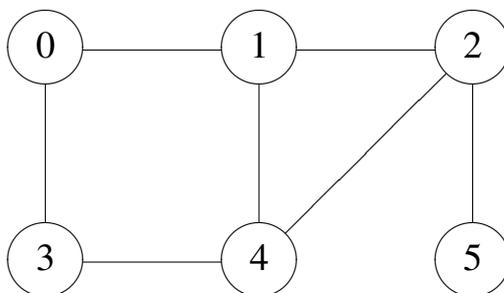
採点プログラムのサンプルが読み込む入力の例と、それに対応する関数の呼び出しの例を以下に示す。

入力例	関数の呼び出しの例	
	呼び出し	戻り値
1	Ask(3, 5, {0,0,1,1,1,1})	1
6	Answer(2, 4)	
7	Answer(2, 5)	
0 1	Answer(3, 4)	
0 3	Ask(0, 4, {1,0,1,0,1,0})	0
1 2	Answer(0, 1)	
1 4	Answer(0, 3)	
2 4	Answer(1, 4)	
2 5	Answer(1, 2)	
3 4		

この例における関数呼び出しは、必ずしも意味のある呼び出しとは限らないことに注意せよ。

この入力例において、関数 Detect は、 $T = 1$ 、 $N = 6$  として呼び出される。

この入力例において、JOI 島は下図に示された構造をしている。



JOI 島の様子。円と数値は広場とその番号を、線分は道を表す。

- 1 回目の関数 Ask の呼び出しでは、広場 2, 3, 4, 5 のみが経由可能であるとき、広場 3 から広場 5 まで移動可能かを尋ねている。移動可能なので、関数 Ask は 1 を返す。
- 2 回目の関数 Ask の呼び出しでは、広場 0, 2, 4 のみが経由可能であるとき、広場 0 から広場 4 まで移動可能かを尋ねている。移動不可能なので、関数 Ask は 0 を返す。