

## 本選会場 (Finals)

JOI 国には  $N$  個の都市があり,  $1$  から  $N$  の番号がついている. また,  $M$  本の道路がある. 全ての道路は異なる 2 つの都市を結び, 双方向に通行可能である. JOI 国の任意の 2 つの都市は, 道路をたどって片方からもう片方へ到達可能である. JOI 国の道路は全て有料道路であり, 道路ごとに通行料金が決まっている.

JOI 国でも情報オリンピックは開催される. JOI 国の情報オリンピックの本選には, 各都市の代表選手が出場する. 本選をどの都市で開催するかを決定し, 選手をそれらの都市に集めるためにかかる金額を見積もっておく必要がある. 本選は  $K$  個の都市で開催し, 本選の際には全ての選手をそれらのいずれかの都市に移動させなければならない. 1 つの都市に集まる選手の人数に制限はない.

本選を開催する都市に選手を集める際には, 道路を利用する. 通行料金が  $c$  の道路は, 一度に何人で通行しても料金は  $c$  である. そこで, 選手を移動させる順番を工夫することによって, 複数の選手を一度に通行させるようにすれば, 料金を節約することができる.

本選を開催する都市を決定し, 選手を本選会場に集める際にかかる通行料金の和を最小化したい.

## 課題 (TASK)

$N, M, K$  と全ての道路の情報が与えられたとき, 選手を本選会場に集める際にかかる通行料金の和の最小値を計算するプログラムを作成せよ.

## 制限 (CONSTRAINTS)

- $1 \leq N \leq 100,000$  都市の数
- $1 \leq M \leq 100,000$  道路の数
- $1 \leq K \leq N$  本選を開催する都市の個数
- $1 \leq A_i < B_i \leq N$  道路  $i$  の結ぶ 2 つの都市
- $1 \leq C_i \leq 100$  道路  $i$  の通行料金

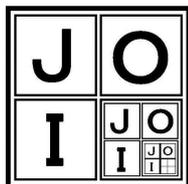
## 入力 (INPUT)

標準入力から以下の入力を読み込め.

- 1 行目には整数  $N, M, K$  が空白を区切りとして書かれている.
- 続く  $M$  行には, 1 行につき 1 つの道路について記述されている. これらの行のうちの  $i$  行目は道路  $i$  について記述しており, 整数  $A_i, B_i, C_i$  が空白を区切りとして書かれている.

## 出力 (OUTPUT)

標準出力に選手を本選会場に集める際にかかる金額の最小値を表す 1 つの整数を出力せよ.



## 採点基準 (GRADING)

40 点分のテストグループにおいて,  $N \leq 1,000$  である.

40 点分のテストグループにおいて,  $K = 1$  である.

20 点分のテストグループは, これら 2 つの条件の両方をみたす.

60 点分のテストグループは, これら 2 つの条件の少なくとも一方をみたす.

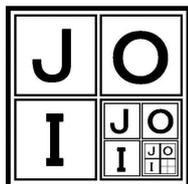
## 入出力例 (EXAMPLE)

入力例 (Sample Input)	出力例 (Sample Output)
4 3 1 1 2 2 2 3 9 2 4 5	16

例えば, 都市 1 で本選を開催するとし, まず都市 4 の代表選手を都市 2 へ移動させ, 次に都市 3 の代表選手を都市 2 へ移動させ, 都市 2 に居る 3 人の選手を都市 1 へ移動させればよい.

入力例 (Sample Input)	出力例 (Sample Output)
5 6 2 1 2 5 1 3 3 2 3 4 2 5 7 3 4 6 4 5 5	12

例えば, 都市 3 と都市 4 で本選を開催するとし, 都市 1, 2 の代表選手は都市 3 へ, 都市 5 の代表選手は都市 4 へ集めればよい.



## かくれんぼ (Hide-and-seek)

あなたは、JOI社が発売したテレビゲームソフトを手に入れた。なかなか良くできたゲームであり、それなりに楽しみながら毎日プレイしていた。

ある日、ゲーマーの中で「かくれんぼ」と呼称されるステージが出現した。どうやらそのステージにはバグがあり、優れたゲーマーでさえもほんのわずかな確率でしかクリアできない物であるらしい。

何度もそのステージに挑戦する中であなたは、とても高速な判断を行うことでクリアできる可能性が有ることに気づき、プログラムを作成して対処出来るのではないかと考えた。

かくれんぼステージは、多数の障害物が配置された場所が舞台となっている。舞台は長方形で  $1 \times 1$  の正方形のマスに分かれており、各マスは  $1 \leq x \leq 100,000$  ,  $1 \leq y \leq 1,000,000,000$  を満たす整数  $x, y$  によって  $(x, y)$  と表現される。(1, 1) は左上隅のマスであり、 $(x + 1, y + 1)$  は (1, 1) から右に  $x$  , 下に  $y$  進んだマスを表示する。

各障害物は  $y$  座標が同じ連続した  $w$  個のマスに置かれる。つまり、障害物は  $w \times 1$  個のマスをおさめる長方形の形に見える。その中の  $x$  座標が一番小さいマスの座標  $(x, y)$  と、長さ  $w$  の組で1つの障害物が表現されている。障害物は  $2 \leq y$  のマスに配置される。障害物同士が重なることはない。

ステージが始まるとプレイヤーは舞台を動き回る。プレイヤーは障害物のあるマスを含めた任意のマスに移動できる。

一定の時間が経過すると敵が出現し攻撃を行う。プレイヤーはこの時必ず障害物の中に隠れる必要がある。障害物の中に隠れるには、障害物のあるマスに居ればよい。適切な障害物の中に隠れることでプレイヤーは攻撃を受けずに済み、敵に反撃するチャンスを得ることが出来る。チャンスを利用することでステージがクリアできる。

敵は  $M$  種類の武器（例えば、拳銃、ライフル、無反動砲、電磁投射砲などなど）を所持している。武器には1から  $M$  の固有の番号がつけられており、 $i$  番の武器には攻撃力  $a_i$  が設定されている。攻撃力は、その数値の分だけ障害物を破壊出来ることを表している。

破壊された障害物の中にプレイヤーが隠れている場合、プレイヤーはダメージを受けることになる。

敵はランダムに選ばれた  $x$  を使い、 $(x, 1)$  に出現し、下方向に向かってランダムに選択した武器を使い攻撃する予定だった。ところが、ゲームのバグによって、敵は必ずプレイヤーのいる  $x$  座標を選択し、プレイヤーに向けて攻撃してしまう事となった。

あなたは自作のプログラムを使い、敵からどの武器を使って攻撃されても良いように、武器毎に攻撃を受けずに済む最適な隠れ場所を探すことにした。攻撃を受けずに済む最適な隠れ場所は、プレイヤーが反撃を行いやすいように、もっとも  $y$  座標が小さい所である。また、そのような場所が複数存在する場合には、その中でもっとも  $x$  座標が小さい所が最適である。

## 課題 (TASK)

障害物の情報と各武器の攻撃力が与えられたとき、敵が所持する武器毎に最適な隠れ場所を求めるプログラムを作成せよ。ただし、どのように隠れても攻撃を受けてしまう場合には、隠れる場所がないという意味で  $(-1, -1)$  を出力せよ。

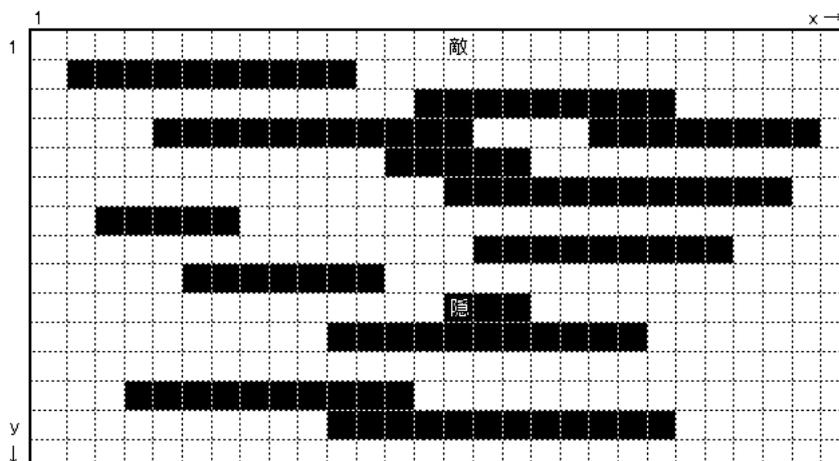


図 1: 攻撃力が 4 である武器に対応する隠れ方

## 制限 (CONSTRAINTS)

- |                                     |                                   |
|-------------------------------------|-----------------------------------|
| $1 \leq N \leq 50,000$              | 障害物の数                             |
| $1 \leq M \leq 50,000$              | 武器の種類                             |
| $1 \leq x_i \leq 100,000$           | 障害物 $i$ が配置されるマスの中でもっとも小さい $x$ 座標 |
| $2 \leq y_i \leq 1,000,000,000$     | 障害物 $i$ の $y$ 座標                  |
| $1 \leq w_i + x_i - 1 \leq 100,000$ | $w_i$ は障害物 $i$ の長さ                |
| $1 \leq a_j \leq N$                 | 武器 $j$ の攻撃力                       |

## 入力 (INPUT)

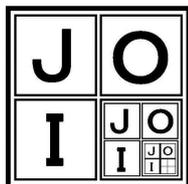
標準入力から以下の入力を読み込め。

- 1 行目には整数  $N$  と  $M$  が空白を区切りとして書かれている。
- 続く  $N$  行のうち  $i$  行目には整数  $x_i, y_i, w_i$  が空白を区切りとして書かれている。
- 続く  $M$  行のうち  $j$  行目には整数  $a_j$  が書かれている。

## 出力 (OUTPUT)

標準出力に以下のデータを出力せよ。

- データは  $M$  行からなる。 $j$  行目は 2 つの整数  $x_j$  と  $y_j$  が空白区切りで書かれており、 $j$  番目の武器に対応する最適な隠れ場所の座標が  $(x_j, y_j)$  である事を表す。どのように隠れても  $j$  番目の武器の攻撃を受けてしまう場合は  $x_j = y_j = -1$  とせよ。



## 採点基準 (GRADING)

30点分のテストグループにおいて,  $x_i + w_i - 1 \leq 10,000$  かつ  $N \leq 1,000$  である.

## 入出力例 (EXAMPLE)

入力例 (Sample Input)	出力例 (Sample Output)
13 2	15 10
2 2 10	-1 -1
14 3 9	
15 6 12	
3 7 5	
16 8 9	
15 10 3	
4 13 10	
11 11 11	
5 4 11	
11 14 12	
6 9 7	
20 4 8	
13 5 5	
4	
7	

この例は図1と対応する. 攻撃力が7の場合は隠れる場所が無いので, -1 -1 を出力する.



## シムロード (SimRoad)

JOI社はシムロードというシミュレーションゲームを販売している。このゲームでは、プレイヤーは舞台となる国の統治者となり、その国を繁栄させるべく様々な操作を行う。

舞台となる国は原始時代である。国には集落が点在しているのだが、この国は多くが背の高い草で覆われており、その状態では集落間を容易に移動することはできない。プレイヤーであるあなたの最初のミッションは住人が集落間を容易に移動できるように草刈をすることである。

国の状態は東西  $W$  マス、南北  $H$  マスで区切られた3種類の状態で表されるマスからなる。'w'はそのマスが草で覆われていることを示し、'.'はそのマスが更地である(草で覆われていない)ことを示し、'@'はそのマスには集落があることを示す。

### 課題 (TASK)

住人は東西南北の隣り合った4方向のマスのうち集落または更地であるマスへは自由に移動することができるが、草で覆われているマスは移動できない。そこで、住人が集落間を容易に移動できるように草で覆われているマスの草を刈り取り更地にすることにより、どの集落からも全ての集落に移動できるように草を刈らなければならない。

草で覆われているマスの草を刈るとそのマスを更地にすることができるが、草刈をするには住人を働かせる必要があり、1マスの草を刈るためには1ドルのお金がかかる。後々の繁栄のことを考えると使うお金は少ない方がよい。

当然、草の刈り取り方は多数存在することがある。与えられた国の状態から、出来る限り草を刈り取るマスが少ない方法で草を刈り、その後の国の状態を生成しなさい。

### 基本アルゴリズム (BASIC ALGORITHM)

$y'$  を  $H$  が偶数であれば  $H/2$  とし、 $H$  が奇数であれば  $(H+1)/2$  とする。まず北から  $y'$  行目のマスのうち草で覆われたマスを全て刈り取り、次に全ての集落について南北方向に北から  $y'$  マス目との間にある草で覆われたマスがあれば全て刈り取る。

### 制限 (CONSTRAINTS)

$1 \leq W \leq 100$  国の東西方向のマスの数

$1 \leq H \leq 100$  国の南北方向のマスの数



## 入力 (INPUT)

入力データは全部で5つあり、ファイル名は `simroad-ink.txt` ( $k = 1, 2, 3, 4, 5$ ) である。入力ファイルは以下のような形式で与えられる。

- 1行目には整数  $W$  と  $H$  が空白を区切りとして書かれている。
- 続く  $H$  行には各行  $W$  文字の文字列が書かれており、国の状態を表す。これらの行の  $y$  行目の  $x$  文字目には、'w' または '.' または '@' の文字が書かれており、北から  $y$  マス目、西から  $x$  マス目のマスの状態を表す。

## 出力 (OUTPUT)

出力データはファイルで提出せよ。出力ファイルのファイル名は `simroad-outk.txt` ( $k = 1, 2, 3, 4, 5$ ) である。 `simroad-outk.txt` には `simroad-ink.txt` に対応した以下の形式のファイルを提出せよ。

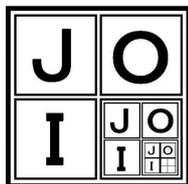
- 草刈をした後の国の状態を書け。出力は  $H$  行からなり、各行は  $W$  文字の文字列である。これらの行の  $y$  行目の  $x$  文字目には、北から  $y$  マス目、西から  $x$  マス目のマスの状態を表す 'w' または '.' または '@' の文字を書け。

## 採点基準 (GRADING)

得点は、あなたが出力した国の状態に依存する。あなたの各々のデータに対する得点は次のように決められる。課題の仕様に合致しない出力である場合、得点は0点となる。仕様に合致している場合は、次のように得点は計算される。 $E_y$  をあなたの出力した国の状態にするためにかかる金額とし、 $E_b$  を基本アルゴリズムが生成する国の状態にするためにかかる金額とし、 $E_m$  を競技参加者が提出した出力の国の状態にするためにかかる金額の最小値とする。この場合のあなたの得点は

- $E_y \geq E_b$  ならば 0
- $E_y < E_b$  ならば  $20 \frac{1/E_y - 1/E_b}{1/E_m - 1/E_b}$

の小数第2位を四捨五入した値である。



## 入出力例 (EXAMPLE)

入力例 (Sample Input)
7 5
w@ww@w@
w.wwwww
www@ww
@wwwww
ww@.@w.

上記の入力例に対する基本アルゴリズムの出力は次のようになり、10マスの草が覆われていたマスの草を刈り取り更地になっているため、10ドルのお金がかかる。

基本アルゴリズムによる出力 (Output by the basic algorithm)
w@ww@w@
w.ww.w.
....@..
@w.w.ww
ww@.@w.

以下は、上記の入力例に対する出力例である。

入力例 (Sample Input)	出力例 (Sample Output)
7 5	w@..@.@
w@ww@w@	w.ww.ww
w.wwwww	www@ww
www@ww	@www.ww
@wwwww	..@.@w.
ww@.@w.	

この出力例の場合は、7マスの草が覆われていたマスの草を刈り取り更地になっているため、7ドルのお金がかかっている。この出力例を提出した場合、もしあなたより小さな金額で作ることのできる国の状態の出力を提出した人がいなければ20点を得るが、 $E_m$ が6ドルであれば12.9点を得ることとなる。