



アジア太平洋情報オリンピック

2010年5月8日（土）

| | 課題 (Task) | | |
|-------------|------------------|--------|-----------|
| | commando | patrol | signaling |
| 時間制限 | 1 秒 | 1 秒 | 2 秒 |
| メモリ制限 | 64 MB | 64 MB | 64 MB |
| 配点 | 100 | 100 | 100 |
| 入力 (Input) | stdin (keyboard) | | |
| 出力 (Output) | stdout (screen) | | |

| 言語 | コンパイラバージョン | コンパイラオプション |
|--------|--------------------|------------|
| C | gcc version 4.1.2 | -m32 -lm |
| C++ | g++ version 4.1.2 | -m32 -lm |
| Pascal | fpc 2.0.4 for i386 | -Sd -Sh |

奇襲部隊 (Commando)

あなたは、1 から n で番号付けられた n 人の兵士からなる軍隊の司令官である。今後の戦いのために、あなたはこの n 人の兵士をいくつかの奇襲部隊に分ける計画を立てる。兵士の結束を強め士気を高めるため、それぞれの部隊は $(i, i+1, \dots, i+k)$ の形の連続する兵士たちからなる。

各兵士 i は戦闘有効度 x_i を持つ。初めは、部隊 $(i, i+1, \dots, i+k)$ の戦闘有効度 x はその部隊に属する兵士の個々の戦闘有効度を足し合わせることで計算されていた。すなわち、 $x = x_i + x_{i+1} + \dots + x_{i+k}$ である。

しかし、あなたは長年の輝かしい勝利の経験から、部隊の戦闘有効度は次のように修正しなければならないと結論付けた：修正後の戦闘有効度 x' は式 $x' = ax^2 + bx + c$ を用いて計算される。ここで、 a, b, c ($a < 0$) は既知の係数で、 x は部隊の初めの戦闘有効度である。

あなたの司令官としての仕事は、全部隊の修正後の戦闘有効度の総和が最大になるように兵士たちをいくつかの部隊に分けることである。

例えば、兵士が4人いて、 $x_1 = 2, x_2 = 2, x_3 = 3, x_4 = 4$ であるとする。さらに、部隊の戦闘有効度を修正する式の係数が $a = -1, b = 10, c = -20$ であるとする。この場合、最適な方法は兵士たちを次の3つの部隊に分けることである：最初の部隊は兵士1と2から、2つ目の部隊は兵士3から、3つ目の部隊は兵士4からなる。3つの部隊の戦闘有効度はそれぞれ4, 3, 4で、修正後の戦闘有効度はそれぞれ4, 1, 4となる。この分け方での戦闘有効度の合計は9で、またこれよりも良い方法は存在しないことは確かめられる。

入力 (Input format)

入力は3行からなる。1行目には兵士の数を表す正整数 n が書かれている。2行目には3つの整数 a, b, c が書かれており、部隊の戦闘有効度を修正する式の係数を表す。3行目には n 個の整数 x_1, x_2, \dots, x_n が空白区切りで書かれており、それぞれ兵士 $1, 2, \dots, n$ の戦闘有効度を表す。

出力 (Output format)

出力は、達成できる修正後の戦闘有効度の最大値を表す整数を含む1行からなる。

入出力例

入力例 (Sample input)

```
4
-1 10 -20
2 2 3 4
```

出力例 (Sample output)

```
9
```

制限 (CONSTRAINTS)

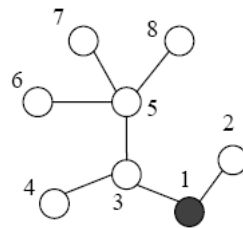
- 配点の 20% 分のテストケースにおいて, $n \leq 1000$ である.
- 配点の 50% 分のテストケースにおいて, $n \leq 10,000$ である.
- 配点の 100% 分のテストケースにおいて, $n \leq 1,000,000$, $-5 \leq a \leq -1$, $|b| \leq 10,000,000$, $|c| \leq 10,000,000$, $1 \leq x_i \leq 100$ である.

パトロール (Patrol)

ある市に、1 から N の番号がついた N 個の集落がある。 $N-1$ 本の道が集落を結んでいて、どの道もちょうど2つの集落を結んでいる。これらの道路を使ってすべての集落はお互いに行き来できる。どの道路も長さは1単位長である。

市民の安全を守るために、市警察の警備隊が全ての道路を回らないとならない。警察署は集落1にあり、その警備隊は集落1から出発し、その日のうちに再び集落1に戻ってくる。

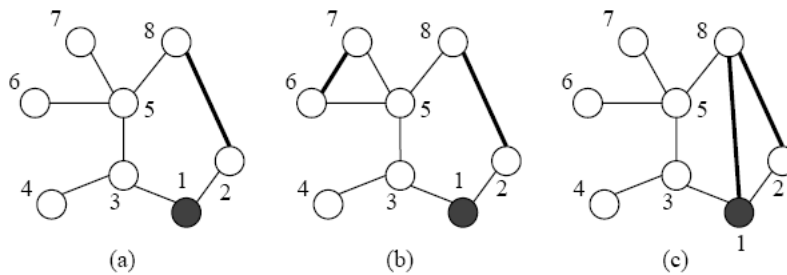
例として下図のような集落が8つある市を考える。集落は円で表されており、集落1は黒い円である。道路はこれらの集落を結ぶ線で表されている。全ての道を通るため、警備隊は毎日14単位長移動する。警備隊は毎日の仕事を完遂するため各道路を2回ずつ通ることに注意せよ。



警備のために移動する距離を減らすため、市は集落を結ぶ K 本の近道を新設することを計画している。近道は2つの集落を結ぶことができる。2つの近道の端点が同じ集落でも良い。(下の例の(c)を見よ。) 近道は同じ集落を結ぶ自己ループでも良い。

財政的な制約から K は1か2である。また、市はお金を無駄にしないために、警備隊はどの近道も毎日ちょうど1回通ることが課されている。

次の例を考える。



例の(a)では、1つの近道が敷設され、移動距離は11になる。例の(b)では、2つの近道が敷設され、移動距離は10になる。例の(c)では、2つの近道が敷設されるが、どの近道もちょうど1回通るという制約のため、移動距離は15になる。

集落間を結んでいる道路の情報と新設する近道の数を読み込むと、警備隊が移動する距離が最小となる近道の敷設場所を計算するプログラムを作成せよ。

入力 (Input format)

入力の 1 行目は 2 つの整数 N と K ($1 \leq K \leq 2$) を含む. 後に続く $N - 1$ 行は道路の情報について書かれている. 各々の行には 2 つの整数 A, B ($1 \leq A, B \leq N$) が書かれており, 集落 A と集落 B を結ぶ道路があることを表している.

出力 (Output format)

K 本の近道を新設した後の警備隊の移動距離の最小値を出力せよ.

入出力例

入力例 1 (Sample input 1)

```
8 1
1 2
3 1
3 4
5 3
7 5
8 5
5 6
```

出力例 1 (Sample output 1)

```
11
```

入力例 2 (Sample input 2)

```
8 2
1 2
3 1
3 4
5 3
7 5
8 5
5 6
```

出力例 2 (Sample output 2)

```
10
```

入力例 3 (Sample input 3)

```
5 2
1 2
2 3
3 4
4 5
```

出力例 3 (Sample output 3)

```
6
```

制限 (CONSTRAINTS)

- 配点の 10% 分のテストケースにおいて、 $N \leq 1,000$ かつ $K = 1$ である。
- 配点の 30% 分のテストケースにおいて、 $K = 1$ である。
- 配点の 80% 分のテストケースにおいて、各集落が隣接する集落数は高々 25 である。
- 配点の 90% 分のテストケースにおいて、各集落が隣接する集落数は高々 150 である。
- 配点の 100% 分のテストケースにおいて、 $3 \leq N \leq 100,000$ かつ $1 \leq K \leq 2$ である。

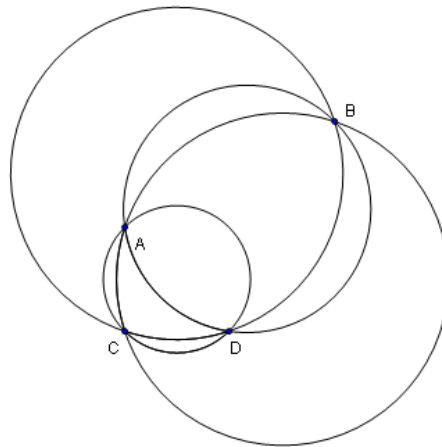
シグナル (Signaling)

ある通信会社が北京市内における GSM ネットワークを開発中である。市内にはネットワークでカバーされる必要がある家が全部で n 個ある。予算の制約により、この会社はアンテナを 1 つだけ設置することができる。

アンテナの配置を簡素化するため、次のようにして設置場所を決める。 n 個の家から 3 個を選び、それらを通る円を描き、アンテナをその円の中心に設置する。このアンテナでカバーされる範囲は円の内部にある全ての家と円周上の家である。

この会社は 3 個の家をランダムに選ぶ予定のため、アンテナの設置場所として全ての可能性を考えたときの、カバーされる家の個数の平均値を求めたい。

例えば、4 個の家 A, B, C, D が以下の図のように位置していたとする。



もし ABC または BCD から定まる円を選んだ場合、全ての家がカバーされる。もし ACD または ABD から定まる円を選んだ場合、4 番目の家はアンテナでカバーされる円の内部に入らない。したがって、カバーされる家の個数の平均値は $\frac{1}{4}(4 + 4 + 3 + 3) = 3.5$ 個である。

あなたの課題は、家の場所が与えられたとき、シグナルでカバーされる家の個数の平均値を計算することである。家の位置は 2 次元の座標系を用いて与えられ、全ての家の座標は整数である。1 つの直線上に 3 つの家が並ぶことはなく、また、1 つの円周上に 4 つの家が並ぶこともないことは保証されている。

入力 (Input format)

入力の 1 行目は家の総数を表す 1 つの正整数 n を含む。続く n 行には家の場所についての情報が書かれている。各 $i \in \{1, 2, \dots, n\}$ に対し、入力の $i+1$ 行目には家 i の座標を表す 2 つの整数 x_i, y_i が空白区切りで書かれている。

出力 (Output format)

シグナルによってカバーされる家の個数の平均値を表す 1 つの実数を出力せよ。出力の絶対誤差は 0.01 以下でなければならない。

入出力例

入力例 (Sample input)

```
4
0 2
4 4
0 0
2 0
```

出力例 (Sample output)

```
3.500
```

例に関する補足

3.5, 3.50, 3.500, ... はいずれも正解とみなされる。また, 3.51, 3.49, 3.499999 など正解とみなされる。

制限 (CONSTRAINTS)

- 配点の 100% 分のテストケースにおいて, 各 $i \in \{1, 2, \dots, n\}$ に対し, 家 i の座標 (x_i, y_i) は両方とも $-1000000 \leq x_i, y_i \leq 1000000$ をみたす整数である。1 つの直線上に 3 つの家が並ぶことはなく, また, 1 つの円周上に 4 つの家が並ぶこともない。
- 配点の 40% 分のテストケースにおいて, $n \leq 100$ である。
- 配点の 70% 分のテストケースにおいて, $n \leq 500$ である。
- 配点の 100% 分のテストケースにおいて, $3 \leq n \leq 1,500$ である。