

D. 家探しゲーム (Guessing Game)

問題名	家探しゲーム (Guessing Game)
実行時間制限	4 sec
メモリ制限	1 GB

ルンドの古い町のある通りには、0 から $N - 1$ までの番号が付いた N 軒の家が一行に並んでいる。Emma はこれらの家のうち 1 つに住んでおり、彼女の友人の Anna と Bertil はどれが Emma の家なのか特定したい。Emma は、自分の家を直接教える代わりに、彼らとゲームをすることに決めた。ゲームを始める前に、Anna と Bertil は通りにある家の数のみを知っている。このとき、Anna と Bertil はある正の整数 K を選び、そして戦略を立てる。これ以降のやりとりは許されない。

このゲームは 2 つのフェーズからなる。1 つ目のフェーズでは、Emma は自分の家が最後になるように、家を訪れる順番を決める。そして Emma は、その順番を事前に Anna に伝えることなく、その順番で Anna を一軒ずつ家に連れていく。Emma の家以外の家に着いたときには、Anna は 1 以上 K 以下の整数をその家のドアにチョークで書く。最後の家、すなわち Emma の家に着いたときには、Emma 自身が家のドアに 1 以上 K 以下の整数を書く。

2 つ目のフェーズでは、Bertil が通りを歩き、Anna や Emma が書いたすべての整数を見る。Bertil はどの家が Emma の家かを推測したい。Bertil には 2 回のチャンスがあり、正しく推測できれば Anna と Bertil の勝ちになる。さもなければ、Emma の勝ちになる。

あなたは Anna と Bertil が必ずこのゲームに勝つことができるような戦略を考案することはできますか？あなたの戦略の得点は K の値によって決められる (小さい方が良い)。

実装

この問題は複数回実行の問題である。すなわち、あなたのプログラムは複数回実行される。1 回目の実行では、Anna の戦略を実行する。その後の実行では Bertil の戦略を実行する。

入力の 1 行目には 2 つの整数 P と N が書かれている。 P は 1 と 2 のいずれかであり、1 つ目のフェーズであるか 2 つ目のフェーズであるかを表す。 N は家の数である。サンプルの入力 (採点には使われない) を除き、 N は常に 100 000 である。

続く入力はフェーズによって変わる:

フェーズ 1

あなたのプログラムはまず K の値を 1 行に出力しなければならない ($1 \leq K \leq 1\,000\,000$). そして, $N - 1$ 回にわたって, 番号 i ($0 \leq i < N$) が書かれた 1 行を読み込み, A_i ($1 \leq A_i \leq K$) を 1 行に出力しなければならない. A_i は Anna が家 i のドアに書いた整数である. Emma の家の番号を除くすべての番号 i は, 採点プログラムが決めた何らかの順番でちょうど 1 回ずつ現れる.

フェーズ 2

あなたのプログラムは N 個の整数 A_0, A_1, \dots, A_{N-1} が書かれた 1 行を読み込まなければならない. A_i は家 i のドアに書かれた整数である.

そして, 2 つの整数 s_1 と s_2 ($0 \leq s_i < N$) を 1 行に出力せよ. これらは Emma の家の番号を推測したものである. s_1 と s_2 は同じでも良い.

実装の詳細

フェーズ 2 に移行するときにプログラムは再スタートされる. これは, 2 つのフェーズを通して情報を何かしらの変数などに保存しておくことができないことを意味する.

各クエリの後には必ず標準出力を flush せよ. さもなくば, あなたのプログラムは Time Limit Exceeded と判定される可能性がある. Python では, `print()` によって自動的に flush される. C++ では, `cout << endl;` によって, 新しい行が出力されるとともに flush がされる. もし `printf` を使う場合は, `fflush(stdout)` を用いよ.

採点プログラムは **アダプティブ** であることがある. すなわち, 採点プログラムはあなたのプログラムの出力に応じて振る舞いを変えることがある. これはヒューリスティックな解法が正解しないようにするためである. 採点プログラムは, フェーズ 1 を試しに実行した後, その実行によって得られた情報をもとに再びフェーズ 1 を実行する可能性がある.

あなたのプログラムは決定的でなければならない. すなわち, 同じ入力に対して 2 回実行したときに同じ振る舞いをしなければならない. もしあなたが乱数をプログラム中に使いたい場合は, 固定されたシード値を用いること.

これは, 事前に決めた定数を `srand` に与える (C++ の場合), または `random.seed` を用いる (Python の場合) ことで実現できる. 別の方法として, C++11 の乱数生成器を使っている場合は, 乱数生成器を構築する際にシード値を指定することでも実現できる. 特に, `srand(time(NULL))` を用いることはできない. もし採点プログラムがあなたのプログラムが決定的でないことを確認した場合には, Wrong Answer と判定される.

もし (最大 3 回の) 実行における実行時間の **総和** が実行時間制限を超過した場合, Time Limit Exceeded と判定される.

評価方法

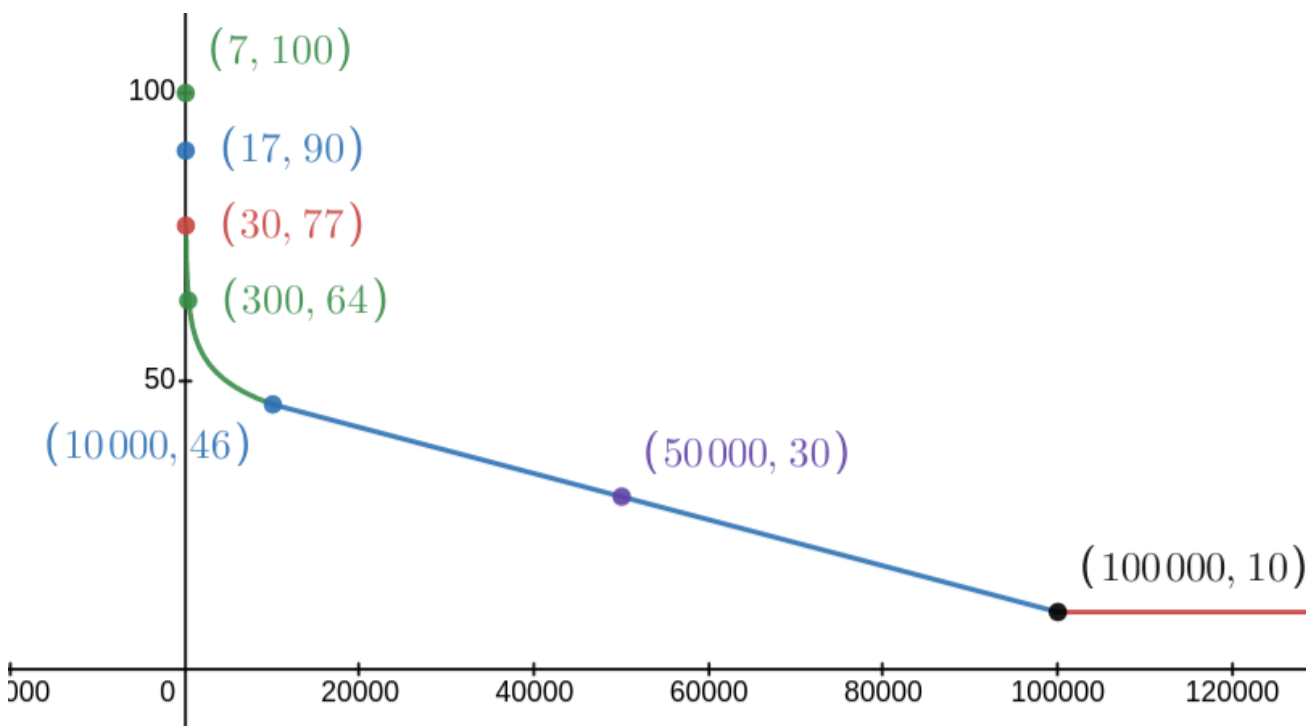
あなたのプログラムはいくつかのテストケースに対してテストされる. もしあなたのプログラムがいずれかのケースで失敗した場合 (答えを間違った場合 (Wrong Answer), 実行時エラー (Runtime Error), 実行時間制限超

過 (Time Limit Exceeded) など), 得点は 0 となり, 適切な判定が下される。

もしあなたのプログラムが すべてのテストケースについて Emma の家の番号を推測できた場合, 判定は Accepted となり, 得点は次のように計算される。 K_{max} をすべてのテストケースに対する K の最大値とする。 K_{max} の値に応じて:

	得点
$K > 99\,998$	10
$99\,998 \geq K > 10\,000$	$10 + \lfloor 40(1 - K/10^5) \rfloor$
$10\,000 \geq K > 30$	$46 + \lfloor 31(4 - \log_{10}(K))/(4 - \log_{10}(30)) \rfloor$
$30 \geq K > 7$	$107 - K$
$7 \geq K$	100

得点を計算するための関数が下に図示されている。



サンプルテストケースは採点には使用されない。そのため, サンプルテストケースについてあなたのプログラムが正しく動作する必要はない。

テストのためのツール

あなたが 解法をテストすることを容易にするため, シンプルなツールがダウンロードできるようになっている。 Kattis の問題ページの下部の "attachments" を見よ。 このツールを必ずしも使う必要はなく, また変更することも許される。ただし, Kattis の実際の採点プログラムとこのツールは異なることに注意せよ。

使用例 ($N = 4$, $s = 2$ の場合. ただし s は最後の家 (Emma の家) のドアに書かれた整数である.):

Python のプログラムの場合, 例えば `solution.py` の場合, (通常は `python3 solution.py` とすれば実行されるが):

```
python3 testing_tool.py python3 solution.py <<<"4 2"
```

C++ のプログラムの場合, まずコンパイルして (例えば `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out` と打って) その後以下を実行せよ:

```
python3 testing_tool.py ./solution.out <<<"4 2"
```

このツールはランダムな順番で家を訪ねる.

特定の順番についてテストしたい場合には, ツールの “MODIFY HERE” と書かれた場所を修正せよ.

例

サンプルテストケースは採点には使用されない. そのため, サンプルテストケースについてあなたのプログラムが正しく動作する必要はない.

$N = 4$ であり, Emma が住む家の番号が 1 であるとする. A をそれぞれの家に書かれた番号のリストであるとする. はじめ, $A = [0, 0, 0, 0]$ である. ただし, 0 はその家に値が何も書かれていないことを表す.

1 回目の実行のとき, あなたのプログラムは:

$N = 4$ が与えられる. あなたのプログラムは $K = 3$ を出力する.

A_2 の値を決めることが要求される. あなたのプログラムは 3 を出力する. A は $[0, 0, 3, 0]$ となる.

A_0 の値を決めることが要求される. あなたのプログラムは 1 を出力する. A は $[1, 0, 3, 0]$ となる.

A_3 の値を決めることが要求される. あなたのプログラムは 2 を出力する. A は $[1, 0, 3, 2]$ となる.

最後に, 採点プログラムが $A_1 = 2$ とする. 最終的に $A = [1, 2, 3, 2]$ となる. これによって, フェーズ 1 が終了する.

2 回目の実行では, あなたのプログラムはリスト 1 2 3 2 を受け取る.

あなたのプログラムは 1 3 を出力する.

推測した番号のうちの 1 つが正しい家の番号 (1) であるため, Anna と Bertil の勝ちとなる.

採点プログラムの出力	あなたのプログラムの出力
1 4	
	3
2	
	3
0	
	1
3	
	2

採点プログラムの出力	あなたのプログラムの出力
2 4	
1 2 3 2	
	1 3