



競走 (Race)

パタヤ市は, IOIと共同で「国際競走オリンピック」 (*International Olympiad in Racing (IOR)*) 2011 を主催することになった. 主催者として, 競走が可能である最適なコースを見つける必要がある.

パタヤ (Pattaya) - チョンブリ (Chonburi) の大都市圏には N 個の都市があり, $N-1$ 本の高速道路網で結ばれている. 各高速道路は双方向に移動可能であり, 異なった都市を結び, その長さはキロメートル単位の整数値をとる. 加えて, どの2つの都市の間も「ちょうど1つ」の経路で結ばれている. 言い換えると, どの都市からどの都市へも, 同じ都市を2度訪れることなく移動することができる方法がちょうど1つ存在する.

IORには, コースの開始都市から終了都市までの経路の合計距離が「ちょうど」 K キロメートルでなければならないという規則がある. 言うまでもなく, 衝突を防ぐためにどの高速道路も(したがって, 都市も), 2度以上コースに使われることはない. 交通の混乱を最小にするために, コースに含まれる高速道路の本数をできる限り少なくする必要がある.

課題 (Your task)

次のパラメータを持つプロシージャ `best_path(N, K, H, L)` を実装せよ:

- N — 都市数. 都市には 0 から $N-1$ までの番号がふられている.
- K — 競走コースの長さ.
- H — 高速道路を表現する2次元配列. $0 \leq i < N-1$ に対し, 高速道路 i は都市 $H[i][0]$ と都市 $H[i][1]$ を結んでいる.
- L — 高速道路の長さを表現する1次元配列. $0 \leq i < N-1$ に対し, 高速道路 i の長さは $L[i]$ である.

配列 H の全ての値が 0 以上 $N-1$ 以下であり, 配列によって表現される高速道路は上記のように全ての都市を結んでいることを仮定して良い. また, 配列 L の全ての値が 0 以上 $1\,000\,000$ 以下であると仮定して良い.

あなたのプロシージャは長さがちょうど K の適切な競走コースの「高速道路の最小本数」を返す必要がある. その様なコースが無い場合は, あなたのプロシージャは -1 を返す必要がある.

例 (Examples)

例 1

例として図 1 に示される $N = 4, K = 3,$

```
0 1    1
H = 1 2  L = 2
    1 3    4
```

の場合を考える.

この例では, コースは都市 0 から開始し, 都市 1 へ向かい, 都市 2 で終了する. このコースの長さはちょうど $1 \text{ km} + 2 \text{ km} = 3 \text{ km}$ であり, 2 本の高速道路を含む. このコースは最適なコースであり, $\text{best_path}(N, K, H, L)$ は 2 を返す必要がある.

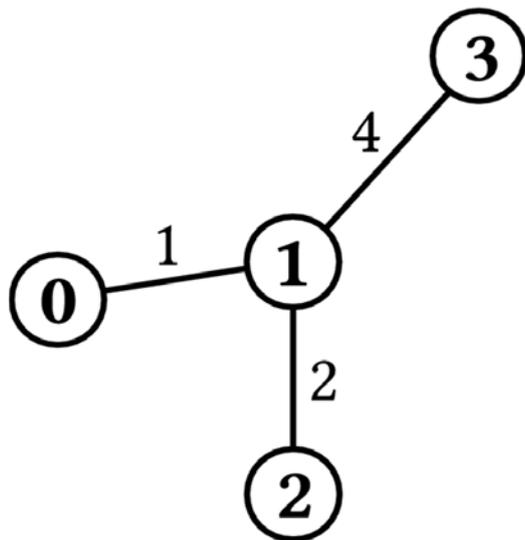


図 1

例 2

例として図 2 に示される $N = 3, K = 3,$

$$H = \begin{matrix} 0 & 1 \\ 1 & 2 \end{matrix} \quad L = \begin{matrix} 1 \\ 1 \end{matrix}$$

の場合を考える.

この例では, 適切なコースは存在しない. この場合 $\text{best_path}(N, K, H, L)$ は -1 を返す必要がある.

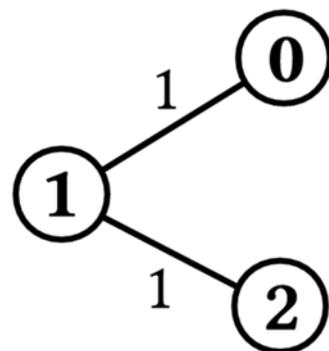


図 2

例 3

例として図 3 に示される $N = 11, K = 12,$

$$H = \begin{matrix} 0 & 1 & 3 \\ 0 & 2 & 4 \\ 2 & 3 & 5 \\ 3 & 4 & 4 \\ 4 & 5 & 6 \\ 0 & 6 & 3 \\ 6 & 7 & 2 \\ 6 & 8 & 5 \\ 8 & 9 & 6 \\ 8 & 10 & 7 \end{matrix} \quad L = \begin{matrix} 3 \\ 4 \\ 5 \\ 4 \\ 6 \\ 3 \\ 2 \\ 5 \\ 6 \\ 7 \end{matrix}$$

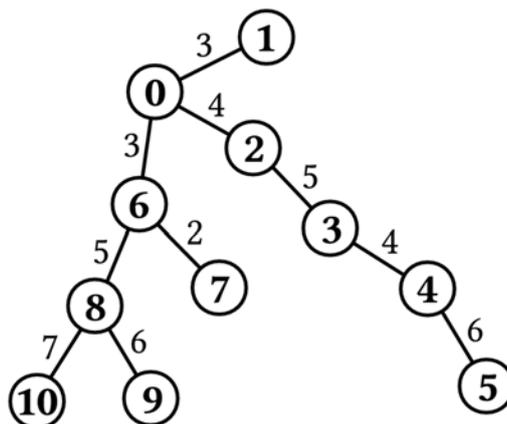


図 3

の場合を考える.

1 つの適切なコースでは, 都市 6 から都市 0 と都市 2 を経由し都市 3 に至る, 3 本の高速道路を含んでいる. 別のコースでは, 都市 10 から開始し都市 8 を経由し都市 6 に至る. 両方のコースの長さは要求通りちょうど 12 km である. 1 本の高速道路からなる適切なコースは存在せず, 2 つ目のコースが最適である. したがって, `best_path(N, K, H, L)` は 2 を返す必要がある.

小課題 (Subtasks)

小課題 1 (9 点)

- $1 \leq N \leq 100$
- $1 \leq K \leq 100$
- 高速道路網は最も単純な直線の形状である: $0 \leq i < N-1$ に対し, 高速道路 i は都市 i と都市 $i+1$ を結んでいる.

小課題 2 (12 点)

- $1 \leq N \leq 1\,000$
- $1 \leq K \leq 1\,000\,000$

小課題 3 (22 点)

- $1 \leq N \leq 200\,000$
- $1 \leq K \leq 100$

小課題 4 (57 点)

- $1 \leq N \leq 200\,000$
- $1 \leq K \leq 1\,000\,000$

実装の詳細 (Implementation details)

制限 (Limits)

- CPU 時間制限 (CPU time limit): 3 秒
 - メモリ制限 (Memory limit): 256 MB
- 注意:** スタックのサイズには決められた制限はない. スタックとして使用されたメモリは, メモリ総使用量に含まれる.

インターフェース (API) (Interface (API))

- 実装フォルダ (Implementation folder): `race/`
 - 選手が実装するファイル: `race.c` または `race.cpp` または `race.pas`
 - 提出ファイルのインターフェース (Contestant interface): `race.h` または `race.pas`
 - 採点プログラムのインターフェース (Grader interface): `race.h` または `racelib.pas`
 - 採点プログラムのサンプル (Sample grader): `grader.c` または `grader.cpp` または `grader.pas`
 - 採点プログラムの入力のサンプル (Sample grader input): `grader.in.1`, `grader.in.2`, ...
- 注意:** 採点プログラムのサンプルは次の書式の入力を読み込む.

- 1 行目: N と K .
- 2 行目から N 行目: 高速道路の情報. つまり, $0 \leq i < N-1$ に対し, $i+2$ 行目には $H[i][0]$ と

$H[i][1]$ と $L[i]$ が空白区切りで書かれている.

- $N+1$ 行目: 期待される解.
- 採点プログラムの入力のサンプルに対して, 期待される出力: `grader.expect.1`, `grader.expect.2`, ...
この課題において, これらのファイルはいずれも文字列 **“Correct.”** のみを含むファイルである.