封鎖時間 (Closing Time)

ハンガリーは N 個の都市からなり，それぞれの都市には 0 から $N-1$ までの番号が付けられている．

これらの都市は $N-1$ 本の *双方向の* 道路で結ばれており，これらの道路には 0 から $N-2$ までの番号が付けられている． $0 \leq j \leq N-2$ である j について，道路 j は都市 $U[j]$ と都市 $V[j]$ を結んでおり，この長さは $W[j]$ である． これは，この道路を通り 2 個の都市間を移動するために $W[j]$ 単位時間を要することを意味する． それぞれの道路は 2 個の異なる街を結んでいる．また，それぞれの都市の組について，多くとも 1 本の道路で結ばれている．

2 個の異なる都市 a と b の間の **パス** とは，異なる都市の列 $p_0, p_1, \ldots, p_t$ であり，以下のようなものである：

- $p_0 = a$,
- $p_t = b$,
- $0 \leq i < t$ である i について，都市 p_i と p_{i+1} を結ぶ道路が存在する．

どの都市からどの都市へもいくつかの道路を使って到達することが可能である．すなわち，どの 2 個の異なる都市についても，パスが存在する． このパスはそれぞれの異なる都市の組について，一意に定まることが示される．

パス $p_0, p_1, \ldots, p_t$ の **長さ** とはパス上で連続した都市を結ぶ t 本の道路の長さの合計である．

ハンガリーでは，2 個の大きな都市で行われる建国記念日の祭りに参加するためにたくさんの人々が移動する． 式典が終わると，人々は各々の家に帰る． ハンガリー政府は混雑によって地元の人々が混乱することを防ぎたい．そのため，政府は各都市について，それぞれを特定の時間に封鎖することを計画している． それぞれの都市は政府によって非負整数である **封鎖時間** が割り当てられる． 政府は封鎖時間の合計を K 以下にすることを決めた． より正確には，0 以上 $N-1$ 以下の i について，都市 i に封鎖時間として非負整数 $c[i]$ を割り当てる． すべての $c[i]$ の合計は K 以下でなければならない．

都市 a といくつかの封鎖時間の割り当てについて検討せよ． 都市 a から都市 b に **到達可能** であるとは，$b = a$ であるとき，あるいはこれらの都市の間のパス $p_0, \ldots, p_t$ （特に $p_0 = a$ かつ $p_t = b$ であるとき）が以下の条件を満たすときである．

- パス p_0, p_1 の長さが $c[p_1]$ 以下であり，かつ
- パス p_0, p_1, p_2 の長さが $c[p_2]$ 以下であり，かつ
- $\ldots$
- パス $p_0, p_1, p_2, \ldots, p_t$ の長さが $c[p_t]$ 以下である．

今年，2 つの祭りは都市 X と Y で開かれる． それぞれの封鎖時間の割り当てについて，**便利さ** は以下の 2 つの値の合計として定義される．

- 都市 X から到達可能な都市の数．
- 都市 Y から到達可能な都市の数．

都市 X から到達可能かつ都市 Y から到達可能な都市について，便利さには 2 回数えられることに注意せよ．

封鎖時間の割り当てにより達成可能な便利さの最大値を求めよ．

実装の詳細

以下の関数を実装せよ．

```
int max_score(int N, int X, int Y, int64 K, int[] U, int[] V, int[] W)
```

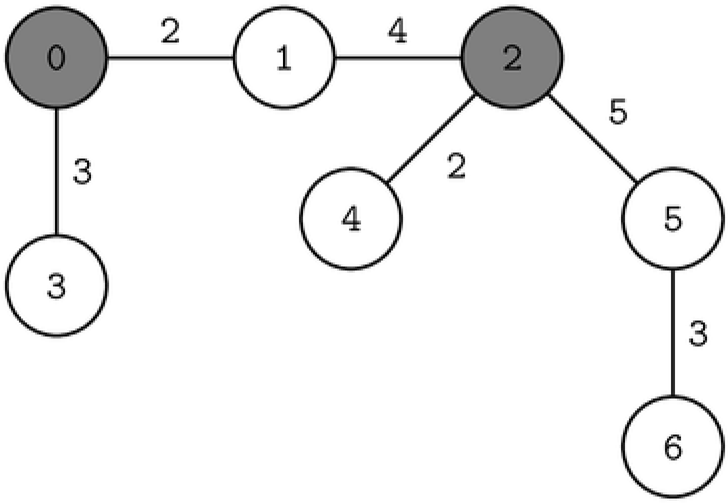
- N：都市の数．
- X，Y：祭りが開かれる都市．
- K：封鎖時間の合計の上限．
- U，V：道路の接続について表す長さ $N-1$ の配列．
- W：道路の長さを表す長さ $N-1$ の配列．
- この関数は封鎖時間の割り当てにより達成可能な便利さの最大値を返さなければならない．
- この関数はそれぞれのテストケースについて **複数回** 呼び出されるかもしれない．

例

以下の呼び出しについて考える：

```
max_score(7, 0, 2, 10,
          [0, 0, 1, 2, 2, 5], [1, 3, 2, 4, 5, 6], [2, 3, 4, 2, 5, 3])
```

これは以下の道路網に対応する：



以下のように封鎖時間を割り当てることを考える：

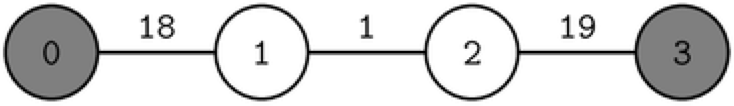
都市	0	1	2	3	4	5	6
封鎖時間	0	4	0	3	2	0	0

封鎖時間の合計は 9であり，$K=10$ 以下であることに注意せよ． 都市 0，1，3 は都市 X（$X=0$）から到達可能であり，都市 1，2，4 は都市 Y（$Y=2$）から到達可能である． したがって，便利さは $3+3=6$ である． 便利さが 6 より大きくなる封鎖時間の割り当て方は存在しないため，この関数は 6 を返さなければならない．

また，以下の呼び出しについて考える：

```
max_score(4, 0, 3, 20, [0, 1, 2], [1, 2, 3], [18, 1, 19])
```

これは以下の道路網に対応する：



以下のように封鎖時間を割り当てることを考える：

都市	0	1	2	3
封鎖時間	0	1	19	0

都市 0 は都市 X（$X=0$）から到達可能であり，都市 2，3 は都市 Y（$Y=3$）から到達可能である． したがって，便利さは $1+2=3$ である． 便利さが 3 より大きくなる封鎖時間の割り当て方は存在しな

いため，この関数は 3 を返さなければならない．

制約

- $2 \le N \le 200\,000$
- $0 \le X < Y < N$
- $0 \le K \le 10^{18}$
- $0 \le U[j] < V[j] < N$ $(0 \le j \le N-2)$
- $1 \le W[j] \le 10^6$ $(0 \le j \le N-2)$
- どの都市からどの都市へもいくつかの道路を使って到達することが可能である．
- $S_N \le 200\,000$ である．ここで，S_N は各テストケース内での max_score のすべての呼び出しにおける引数 N の和である．

小課題

ここで，道路網が **直線状** であるとは，（$0 \le i \le N-2$ である i について）道路 i が都市 i と都市 $i+1$ を結んでいることを指す．

1. （8 点） 都市 X と都市 Y を結ぶパスの長さは $2K$ よりも長い．
2. （9 点） $S_N \le 50$，道路網は直線状である．
3. （12 点） $S_N \le 500$，道路網は直線状である．
4. （14 点） $S_N \le 3\,000$，道路網は直線状である．
5. （9 点） $S_N \le 20$
6. （11 点） $S_N \le 100$
7. （10 点） $S_N \le 500$
8. （10 点） $S_N \le 3\,000$
9. （17 点） 追加の制約はない．

採点プログラムのサンプル

C 個のシナリオがあるとする．これは，max_score が呼ばれる回数を表す． 採点プログラムのサンプルは以下の形式で入力を読み込む：

- 1 行目：C

これは以下に C 個のシナリオが続くことを表す．

採点プログラムのサンプルはそれぞれのシナリオについて以下の形式で入力を読み込む：

- 1 行目：$N\ X\ Y\ K$
- $2+j$ 行目 $(0 \le j \le N-2)$：$U[j]\ V[j]\ W[j]$

採点プログラムのサンプルは以下の形式でそれぞれのシナリオについて 1 行で出力する：

- 1 行目：max_score の戻り値