

通行止め (Road Closures)

インドネシアの大都市スラバヤには N 個の交差点があり、 0 から $N - 1$ までの番号が付けられている。これらの交差点は、 $N - 1$ 本の双方向に通行可能な道路によってひとつながりになっており、道路には 0 から $N - 2$ までの番号が付けられている。どの 2 つの交差点についても、道路をたどって行き来する経路がただ 1 つ存在する。道路 i ($0 \leq i \leq N - 2$) は交差点 $U[i]$ と交差点 $V[i]$ をつないでいる。

スラバヤの市長である Pak Dengklek は、地球環境に対する市民の意識を高めるために、車のない日 (Car Free Day) というイベントを計画している。このイベントを奨励するため、Pak Dengklek は、最初に非負整数 k を選んだ後、いくつかの道路を通行止めにする。この際に、どの交差点についても、これに接続する通行止めになっていない道路が k 本以下 になっている必要がある。道路 i を通行止めにするには $W[i]$ のコストがかかる。

Pak Dengklek を助けるために、それぞれの k ($0 \leq k \leq N - 1$) の選び方について、通行止めにかかるコストの合計の最小値を求めよ。

実装上の注意 (Implementation Details)

以下の関数を実装せよ。

```
int64[] minimum_closure_costs(int N, int[] U, int[] V, int[] W)
```

- N : スラバヤにある交差点の個数。
- U, V : それぞれ長さ $N - 1$ の配列であり、道路 i が交差点 $U[i]$ と交差点 $V[i]$ をつないでいることを意味する。
- W : 長さ $N - 1$ の配列であり、 $W[i]$ は道路 i を通行止めにするのにかかるコストを表す。
- この関数は長さ N の配列を返さなければならない。それぞれの k ($0 \leq k \leq N - 1$) について、戻り値の k 番目の要素は、どの交差点についてもこれに接続する通行止めでない道路が k 本以下であるための、道路の通行止めにかかるコストの合計の最小値でなければならない。
- この関数は 1 回だけ呼び出される。

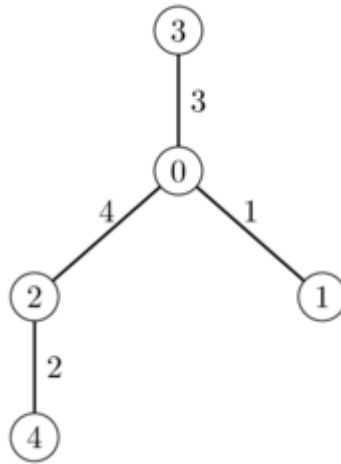
入出力例 (Examples)

例 1

以下の関数呼び出しを考える。

```
minimum_closure_costs(5, [0, 0, 0, 2], [1, 2, 3, 4], [1, 4, 3, 2])
```

この例では、スラバヤには 5 個の交差点と 4 本の道路がある。各道路がつなぐ交差点の組は $(0,1)$, $(0,2)$, $(0,3)$, $(2,4)$ である。通行止めにかかるコストは、それぞれ 1, 4, 3, 2 である。



コストを最小にするために：

- $k = 0$ を選ぶ場合、すべての道路を通行止めにする。コストの合計は $1 + 4 + 3 + 2 = 10$ である。
- $k = 1$ を選ぶ場合、道路 0 と道路 1 を通行止めにする。コストの合計は $1 + 4 = 5$ である。
- $k = 2$ を選ぶ場合、道路 0 を通行止めにする。合計で 1 のコストがかかる。
- $k = 3$ または $k = 4$ を選ぶ場合、道路をひとつも通行止めにしなない。

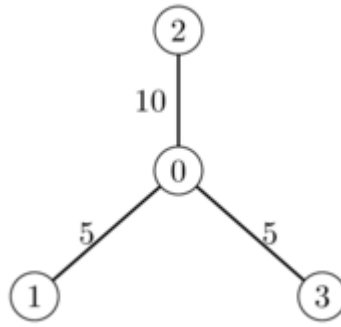
すなわち、関数 `minimum_closure_costs` は $[10, 5, 1, 0, 0]$ を返さなければならない。

例 2

以下の関数呼び出しを考える。

```
minimum_closure_costs(4, [0, 2, 0], [1, 0, 3], [5, 10, 5])
```

この例では、スラバヤには 4 個の交差点と 3 本の道路がある。各道路がつなぐ交差点の組は $(0,1)$, $(2,0)$, $(0,3)$ である。通行止めにかかるコストは、それぞれ 5, 10, 5 である。



コストを最小にするために：

- $k = 0$ を選ぶ場合，すべての道路を通行止めにする．コストの合計は $5 + 10 + 5 = 20$ である．
- $k = 1$ を選ぶ場合，道路 0 と道路 2 を通行止めにする．コストの合計は $5 + 5 = 10$ である．
- $k = 2$ を選ぶ場合，道路 0 または道路 2 のどちらか一方を通行止めにする．いずれの場合でも合計で 5 のコストがかかる．
- $k = 3$ を選ぶ場合，道路をひとつも通行止めにしない．

すなわち，関数 `minimum_closure_costs` は $[20, 10, 5, 0]$ を返さなければならない．

制約 (Constraints)

- $2 \leq N \leq 100\,000$.
- $0 \leq U[i], V[i] \leq N - 1$ ($0 \leq i \leq N - 2$).
- どの 2 つの交差点についても，いくつかの道路をたどって行き来できる．
- $1 \leq W[i] \leq 10^9$ ($0 \leq i \leq N - 2$).

小課題 (Subtasks)

1. (5 点) $U[i] = 0$ ($0 \leq i \leq N - 2$).
2. (7 点) $U[i] = i, V[i] = i + 1$ ($0 \leq i \leq N - 2$).
3. (14 点) $N \leq 200$.
4. (10 点) $N \leq 2000$.
5. (17 点) $W[i] = 1$ ($0 \leq i \leq N - 2$).
6. (25 点) $W[i] \leq 10$ ($0 \leq i \leq N - 2$).
7. (22 点) 追加の制約はない．

採点プログラムのサンプル (Sample Grader)

採点プログラムのサンプルは，入力を以下の形式で読み込む．

- 1 行目: N
- $2 + i$ 行目 ($0 \leq i \leq N - 2$): $U[i] \ V[i] \ W[i]$

採点プログラムのサンプルは，関数 `minimum_closure_costs` で返された配列を 1 行で出力する．