# Zadanie: BUD
# Najwspólniejszy podciag

**Warsztaty ILO 2017-2018, grupa olimpijska, dzień 8. Dostępna pamięć: 256 MB.**

## Rozwiązanie wzorcowe $O(n + q \cdot log(n))$

Na początek rozpatrzmy nie graf, a drzewo o $n$ wierzchołkach ukorzenione w jakimś wierzchołku. Zdefiniujmy tablice $up[]$ oraz $down[]$, gdzie $up[i]$ jest liczbą zapytań, w trakcie których przejechaliśmy $i$-tą krawędzią w góre drzewa. Analogicznie definiujemy $down[i]$.
Dla każdego zapytania (u, v) istnieje jednoznacznie wyznaczona ścieżka z u do v. Zdefiniujmy $x = LCA(u, v)$ (najniższy wspólny przodek wierzchołków u i v). Wtedy dla każdej krawędzi od $u$ do $x$ zwiększamy $up[kraw]$ a z $x$ do $u$ zwiększamy $down[kraw]$, wtedy kosztem dla danej krawędzi jest $min(up[kraw], down[kraw]) \cdot c_k raw$, a końcowy wynik to suma kosztów wszystkich krawędzi. Jednak ten pomysł działa w złożoności $O(n)$ (nie uwzględniając lca), co jest nieco za wolne.
Postarajmy się przyspieszyć poprzedni pomysł. Zdefiniujmy dwie dodatkowe tablice $upHelper[]$ i $downHelper[]$. Chcemy, żeby $up[kraw] = \sum upHelper[v]$ gdzie $v$ jest wierzchołkiem z poddrzewa poniżej aktualnej krawędzi. Analogicznie dla $downHelper[]$ i $down[]$. Rozpatrzmy zapytanie (u, v) oraz zdefiniujmy $x = LCA(u, v)$. Aby otrzymać odpowiednie wartości w naszych tablicach, musimy ustawić $upHelper[u] + +, downHelper[v] + +, upHelper[x] - -, downHelper[x] - -$. Jesteśmy w stanie zrobić to $O(log(n))$. Wtedy jak będziemy szli od liści do korzenia drzewa otrzymamy odpowiednie wartości po jednym przejściu po drzewie.

Mamy rozwiązanie działające w odpowiedniej złożoności, ale działające na drzewach, a nie na naszym grafie. Postarajmy się zmienić graf na drzewo.

**Obserwacja.1.** *Jedyne ważne krawędzie, to mosty*

*Dowód.* Jest tak, ponieważ pozostaną spójne, które te mosty łączą. W każdej takiej spójnej możemy utworzyć silną spójną składową za pomocą skierowań, więc przejdziemy z każdego wierzchołka do każdego innego bez żadnych kosztów. □

Możemy znaleźć wszystkie mosty za pomocą funkcji LOW i ze spójnych, które te mosty łączą, zbudować drzewo i zastosować na nim wcześniej opisany algorytm.
Złozoność znalezienia mostów i stworzenia drzewa jest liniowa względem wielkości grafu, natomiast wcześniej zastosowany algorytm ma złożoność $O(q \cdot log(n))$, więc końcowa złożoność to $O(n + q \cdot log(n))$.