Anna Pacanowska, 306412

Niech $D[w]$ oznacza długosć najkrótszej ścieżki z wierzchołka w do wierzchołka v, a L zawiera wierzchołki posortowane według ich odległości od v (tablicy D). W ADJ[v] trzymamy listę sąsiadów v.

Aby obliczyć liczbę sensownych dróg z u do v, wykorzystamy następujący algorytm:

1. Oblicz tablicę D, korzystając algorytmu Dijkstry.
2. Oblicz tablicę L, sortując wierzchołki według ich wartości w tablicy D w czasie $n \log n$.
3. $path[v] = 1$
4. for $i \in 2, 3, ..., n$:
    $x = L[i]$
    for $y \in ADJ[x]$:
        if $D[x] > D[y]$:
            $path[x]{+}{=}path[y]$
5. Wynikiem jest $path[u]$.

Zauważmy, że droga $u_1, u_2, ..., u_{k-1}, u_k$ jest sensowna wtedy i tylko wtedy, gdy dla każdego $i \in \{2, 3, ..., k\}$ $D[u_{i-1}] > D[u_i]$.

Poprawność algorytmu udowodnimy przez indukcję po odległości od wierzchołka v. Dla v istnieje dokładnie jedna sensowna ścieżka - pusta. To zaznaczamy w kroku 3.

Weźmy dowolny wierzchołek w. Załóżmy, że liczba sensownych dróg dla wierzchołków bliższych v jest już poprawnie obliczona. Każda sensowna droga z w jest w postaci $w, u_1, u_2, ..., v$, gdzie $D[u_1] < D[u_w]$, a $u_1, u_2, ..., v$ jest sensowna. Stąd liczba takich dróg jest sumą liczb dróg z każdego z wierzchołków bliższych v. Z założenia indukcyjnego wiemy, że te wartości są już poprawnie obliczone, zatem liczba sensownych dróg z wierzchołka w również zostanie poprawnie obliczona.

Teraz pokażemy złożoność obliczeniową podanego algorytmu. Kroki 1. oraz 2. zostaną wykonane w czasie $O(n \log n)$. Pętla po i zostanie wykonana $O(n)$ razy. Pętla po k sumarycznie zostanie wykonana dwa razy dla każdej krawędzi, a więc $O(m)$. Stąd krok 4. zajmie czas $O(n + m)$. Cały algorytm zostanie więc wykonany w czasie $O(m + n \log n)$.