

# Lista 8

## Zadanie 6

---

### 1 Problem

Rozważamy graf skierowany z wagami na krawędziach (możliwe ujemne wagi). Chcemy uzyskać algorytm do wyznaczania najkrótszej ścieżki pomiędzy dowolnymi dwoma wierzchołkami przy czym taka ścieżka nie może być dłuższa niż  $k$ .

### 2 Concept

Użyjemy tutaj zmodyfikowanej wersji algorytmu Bellmana-Forda. Otóż, w oryginalnej wersji tego algorytmu rozważamy drogi o długości ograniczonej jedynie przez liczbę wierzchołków w danym grafie. Takie ograniczenie dawało nam złożoność obliczeniową  $O(|E| \cdot |V|)$  jako, że wszystkie krawędzie skanowane są w najgorszym przypadku  $|V| - 1$  razy. Tutaj interesują nas ścieżki o długości maksymalnie  $k$ .

### 3 Rozwiązanie

W poniższym algorytmie używamy zmodyfikowanej wersji algorytmu Bellmana-Forda. Wprowadzamy dodatkowe ograniczenie w postaci maksymalnej liczby iteracji głównej pętli. (Wierzchołkiem startowym jest  $s$ .)

---

**Algorithm 1** Zmodyfikowany algorytm Bellmana-Forda

---

```
1: for all  $v \in V$  do:
2:    $v.\text{dist} \leftarrow \infty$ 
3:    $v.\text{prev} \leftarrow \text{NULL}$ 
4: end for
5:  $s.\text{dist} \leftarrow 0$ 
6:  $s.\text{prev} \leftarrow s$ 
7:  $\text{changed} \leftarrow \text{TRUE}$ 
8:  $i \leftarrow 0$ 
9: repeat:
10:   $i \leftarrow i + 1$ 
11:   $\text{changed} \leftarrow \text{FALSE}$ 
12:  for all  $(u, v) \in E$  do:
13:    if  $v.\text{dist} > u.\text{dist} + c(u, v)$  then:
14:       $v.\text{dist} \leftarrow u.\text{dist} + c(u, v)$ 
15:       $v.\text{prev} \leftarrow u$ 
16:       $\text{changed} \leftarrow \text{TRUE}$ 
17:    end if
18:  end for
19: until  $\text{changed} \equiv \text{FALSE} \vee i = k$ 
```

---

W linii 19 został dodany dodatkowy warunek końcowy pętli. Ogranicza to maksymalną długość ścieżki w grafie oraz zmienia złożoność obliczeniową algorytmu, która teraz wynosi  $O(k \cdot |E|)$ .

Długość żądanej najkrótszej ścieżki oczywiście mieści się w parametrze „dist” wierzchołka docelowego. Żeby uzyskać drogę „wierzchołek po wierzchołku” należy iteracyjnie przejść po wierzchołkach znalezionej drogi używając parametru „prev” aż nie dojdziemy do wierzchołka startowego.