

## 1 Lista 8, Zadanie 2

### 1.1 Wstęp

Mamy wykonać podobną pracę co w zadaniu 1. Mianowicie:

- wypisz w tabeli dystans do każdego wężła grafu po każdej iteracji algorytmu,
- narysuj drzewo najkrótszych ścieżek powstałe po wykonaniu algorytmu.

Przy czym wykonamy to dla poniższego grafu:

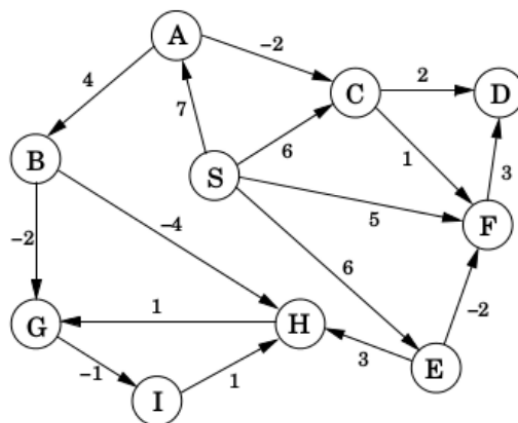


Figure 1: Graf do przeanalizowania

Oraz zamiast używać algorytmu Dijkstry, użyjemy **algorytmu Bellmana-Forda**, ponieważ występują ujemne wagi krawędzi.

## 1.2 Algorytm Bellmana-Forda

```
for  $v \in V$  do
     $v.dist = \infty$ 
     $v.prev = null$ 
end
 $s.dist = 0$ 
 $s.prev = s$ 
while  $change == true$  do
     $change = false$ 
    for all  $(u, v) \in E$  do
        if  $v.dist > u.dist + l(u, v)$  then
             $v.dist = u.dist + l(u, v)$ 
             $v.prev = u$ 
             $change = true$ 
        end
    end
end
end
```

Powyższy algorytm pochodzi z wykładu dr. Gołębiewskiego. Algorytm z "Wprowadzanie do Algorytmów" Cormena (24.1, Bellman-Ford) wykonuje się  $|V| - 1$  razy, co dałoby nam 9 kolumn w analizie, natomiast algorytm z wykładu da zaledwie 5. Będzie czytelniej i przyjemniej.

Przy czym ilość iteracji pętli while zależy głównie od kolejności rozpatrywania krawędzi (w naszym pseudokodzie stosujemy  $(u, v) \in E$ ). Bez wcześniejszej znajomości wag, kolejność może być dowolna. Przyjętą przeze mnie kolejność, zaznaczę neonowo różowym kolorem, gdzie  $i$  to kolejność.

## 1.3 Analiza

Przerysowywanie grafu, co iterację byłoby karkołomne i z jakiegoś powodu kojarzy mi się z podróżą do Mordoru, także zmiany wartości *dist* oraz *prev* węzłów grafu przedstawię w tabeli. Przyjmiemy oznaczenie, że krotka ( $a : b$ ) w komórce tabeli, odpowiada  $a$  - prev,  $b$  - dist.

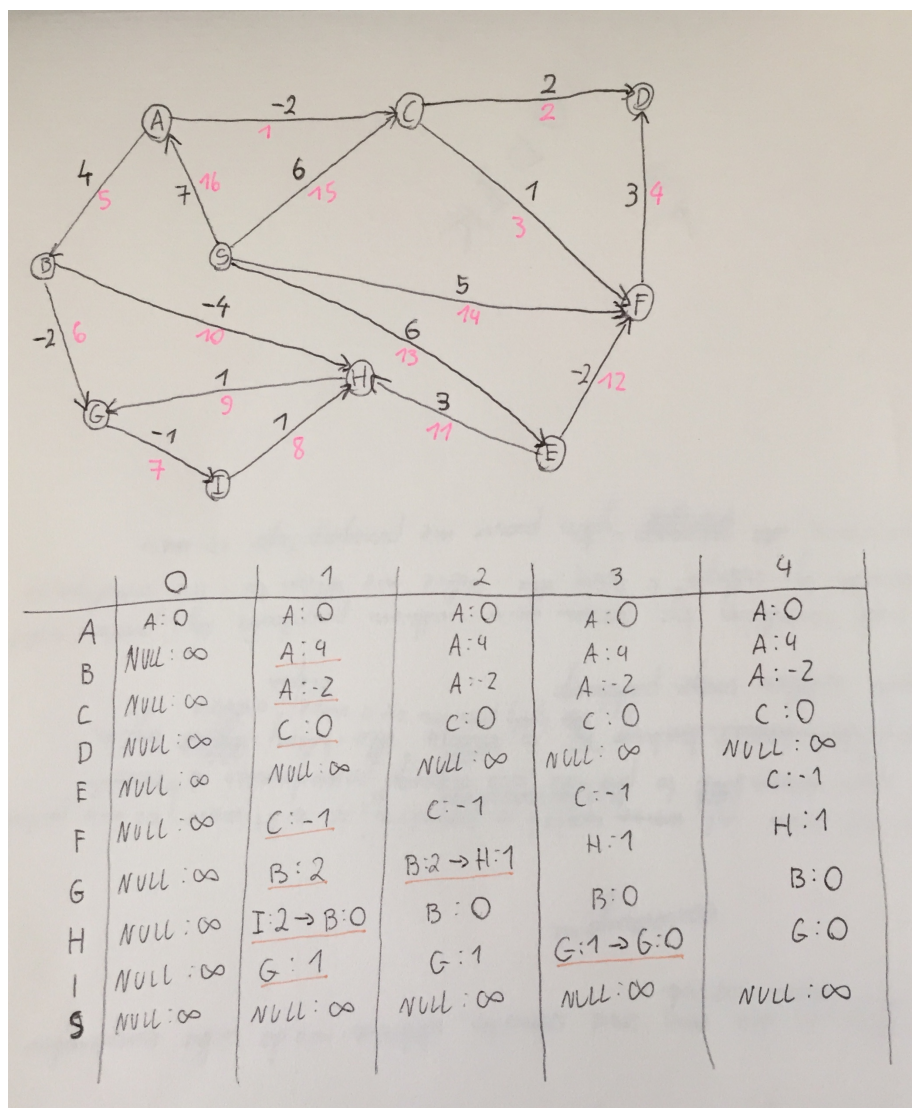


Figure 2: Rozwiązanie pierwszej kropki

Drzewo najkrótszych ścieżek jest jednoznacznie wyznaczone przez  $a$  w czwartej kolumnie tabeli:

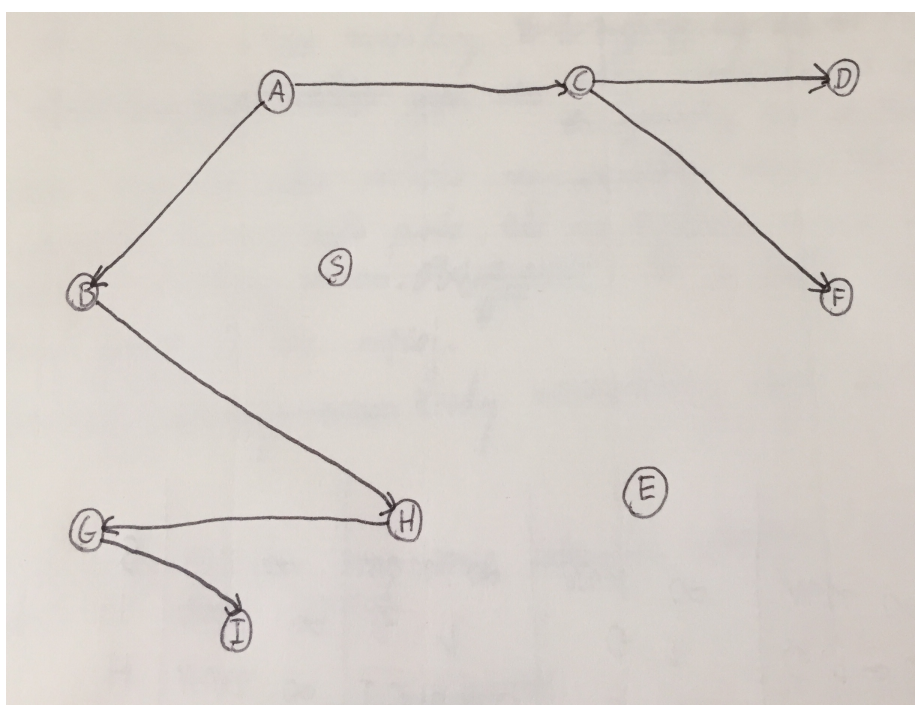


Figure 3: Rozwiązanie drugiej kropki