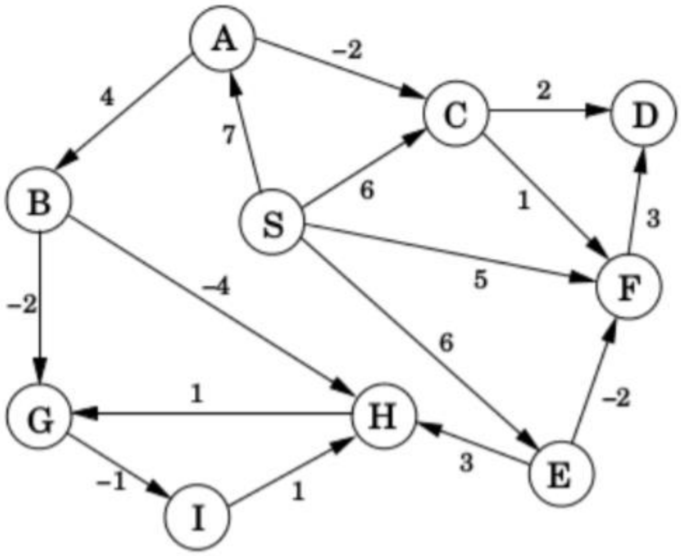
Lista 8 Zad 2

„Wykonaj podobną pracę jak w zadaniu 1 dla poniższego grafu. Jaki algorytm wybierzesz do wyznaczenia najkrótszych ścieżek od wierzchołka A?”



Algorytm:

Do rozwiązania tego zadania zostanie użyty algorytm Bellmana-Forda, który został przedstawiony na wykładzie:

```
Bellman-Ford(G, s)
        for all v \in V
                v.dist = \infty
                v.prev = null
        s.dist = 0
        s.prev = s

        repeat until change == false
                change = false
                for all (u,v) \in E
                        if v.dist > u.dist + l(u,v)
                                v.dist = u.dist + l(u,v)
                                v.prev = u
                                change = true
```

Złożoność obliczeniowa tego algorytmu jest liniowa i wynosi $O(|V| * |E|)$.

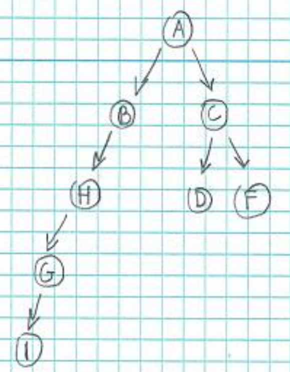
Ów algorytm, w przeciwieństwie do algorytmu Dijkstry, działa poprawnie dla krawędzi o ujemnych wagach.

- Po dokładnej analizie grafu można dojść do wniosku, że do wierzchołka **S** w żaden sposób nie można dojść, a co za tym idzie wierzchołek **E** nie będzie rozpatrywany (wierzchołki te są izolowane).
  Dlatego w pierwszej części rozwiązania rozpatrzymy graf bez wierzchołków **S** i **E,** startując z wierzchołka **A**.
  Natomiast druga część rozwiązania rozpatrzy graf z polecenia jednak zaczniemy z **S** jako wierzchołek startowy.

Start z wierzchołka A (krawędzie S i E są nieosiągalne):

- W pierwszej iteracji algorytmu wierzchołek startowy otrzymuje wartość 0, natomiast reszta wierzchołków otrzymuje wartość ∞.
- Algorytm przechodzi po każdej krawędzi {v, u}, przypisując wartości, wartość v + waga krawędzi oraz nazwę wierzchołka v ⇔ wartość v + waga krawędzi jest mniejsza od wartości u, w przeciwnym wypadku wartość u oraz nazwa wierzchołka-rodzica jest niezmieniana.
- Gdy w danej iteracji nie nastąpi żadna zmiana wartości wierzchołka, algorytm kończy działanie.
- Na podstawie nazw wierzchołków-rodziców, przypisanych do każdego wierzchołka (oprócz roota i wierzchołków izolowanych), można wyznaczyć drzewo rozpinające grafu.

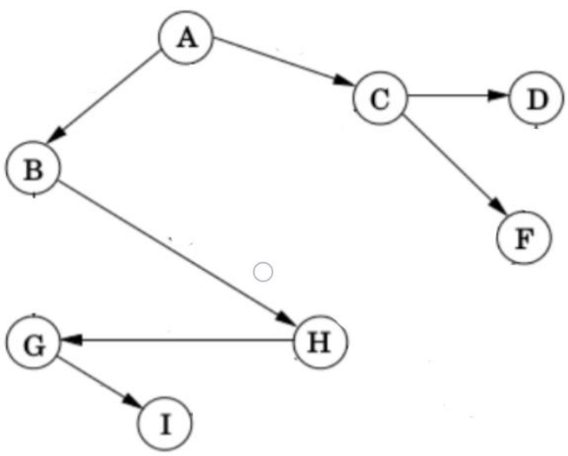
WIERZCHOŁEK \ ITERACJA	0	1	2	3
A	0	0	0	0
B	∞	4(A)	4(A)	4(A)
G	∞	~~2(B)~~ 1(H)	1(H)	1(H)
I	∞	1(G)	0(G)	0(G)
H	∞	~~2(I)~~ 0(B)	0(B)	0(B)
C	∞	-2(A)	-2(A)	-2(A)
D	∞	0(C)	0(C)	0(C)
F	∞	-1(C)	-1(C)	-1(C)
S	∞	∞	∞	∞
E	∞	∞	∞	∞



KRAWĘDZIE: (AB) (BG) (GI) (IH) (HG) (BH) (AC) (CD) (CF) (FD)
(SA) (SC) (SF) (SE) (EF) (EH)

■ - NOWE WARTOŚCI

Drzewo rozpinające, startując od A:

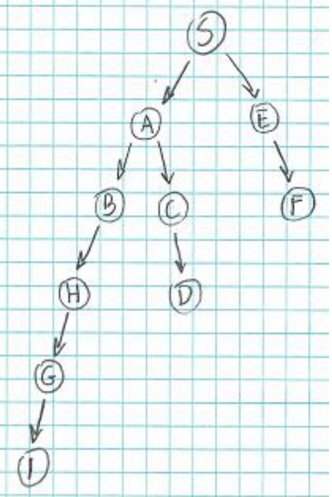


Start z wierzchołka S:

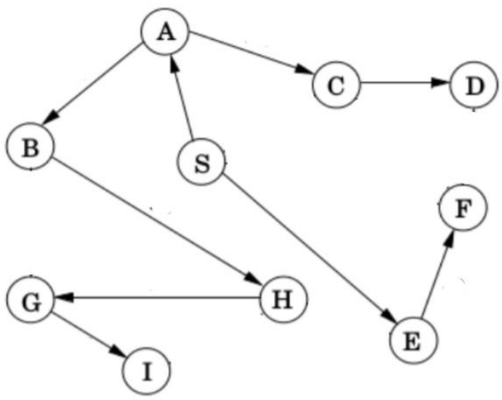
WIERZCHOŁEK \ ITERACJA	0	1	2	3	
S	0	0	0	0	0
A	∞	7(S)	7(S)	7(S)	7(S)
B	∞	11(A)	11(A)	11(A)	11(A)
G	∞	9(B)	~~9(B)~~ 8(H)	8(H)	8(H)
I	∞	8(G)	8(G)	7(G)	7(G)
H	∞	~~9(I)~~ 7(B)	7(B)	7(B)	7(B)
C	∞	5(A)	5(A)	5(A)	5(A)
D	∞	7(C)	7(C)	7(C)	7(C)
F	∞	~~6(D)~~ ~~5(S)~~ 4(E)	4(E)	4(E)	4(E)
E	∞	6(S)	6(S)	6(S)	6(S)

KRAWĘDZIE : (SA) (AB) (BG)(GI)(IH)(HG)(BH) (AC)(SC)
(CD) (CF) (FD)(SF)(SE)(EF)(EH)

- NOWE WARTOŚCI



Drzewo rozpinające, startując od S:



Należy pamiętać, że algorytm Bellmana-Forda wykona maksymalnie |V|-1 iteracji. W ostatniej iteracji żadna z wartości przy wierzchołku nie powinna się zmienić. Jeśli tak się stanie, powodem tego może być **cykl ujemny** zawarty w grafie**,** na który algorytm Bellmana-Forda jest nie odporny. Cykl ujemny jest wtedy, gdy suma wag wszystkich krawędzi cyklu jest ujemna.