



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content
pentru învățământul superior tehnic

Proiectarea Algoritmilor

17. Algoritmul Bellman-Ford

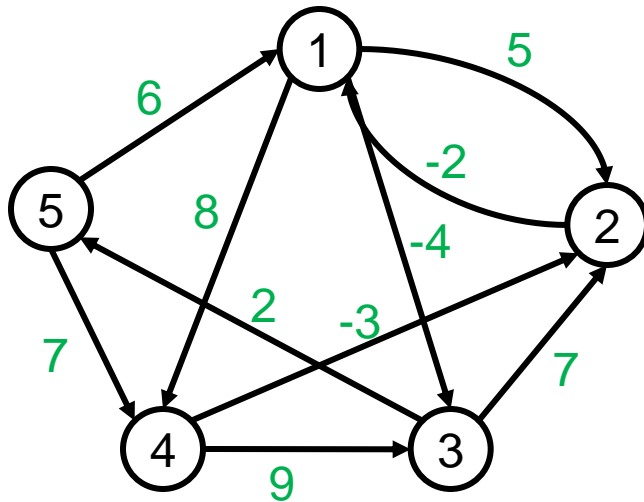
Bibliografie

- [1] R. Sedgewick, K. Wayne - Algorithms and Data Structures Fall 2007 – Curs Princeton - <http://www.cs.princeton.edu/~rs/AlgsDS07/>
- [2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*,

Algoritmul Bellman-Ford

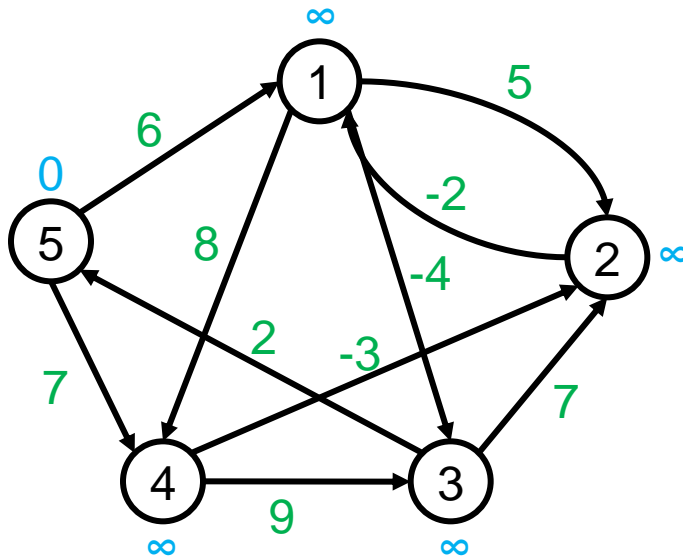
- BellmanFord(G,s) // $G=(V,E),s=sursa$
 - Pentru fiecare v din V // inițializări
 - $d[v] = \infty$;
 - $p[v] = null$;
 - $d[s] = 0$; // actualizare distanță de la s la s
 - Pentru i de la 1 la $|V| - 1$ // pentru fiecare pas de la s spre $V-s$
 - Pentru fiecare (u,v) din E // pentru arcele ce pleacă de la nodurile // deja considerate
 - Dacă $d[v] > d[u] + w(u,v)$ atunci // se relaxează arcele corespunzătoare
 - $d[v] = d[u] + w(u,v)$;
 - $p[v] = u$;
 - Pentru fiecare (u,v) din E
 - Dacă $d[v] > d[u] + w(u,v)$ atunci
 - Eroare ("ciclu negativ");

Exemplu Bellman-Ford (I)



- BellmanFord(G,s)
 - Pentru fiecare v din V // init
 - $d[v] = \infty$;
 - $p[v] = \text{null}$;
 - $d[s] = 0$; // actualizare distanță până la s
 - Pentru i de la 1 la $|V| - 1$ // pt // fiecare pas de la s spre $V-s$
 - Pentru fiecare (u,v) din E // pt. // arcele ce pleacă de la nodurile // deja considerate
 - Dacă $d[v] > d[u] + w(u,v)$ atunci // se relaxează arcele // corespunzătoare
 - $d[v] = d[u] + w(u,v)$;
 - $p[v] = u$;
 - Pentru fiecare (u,v) din E
 - Dacă $d[v] > d[u] + w(u,v)$ atunci
 - Eroare ("ciclu negativ");

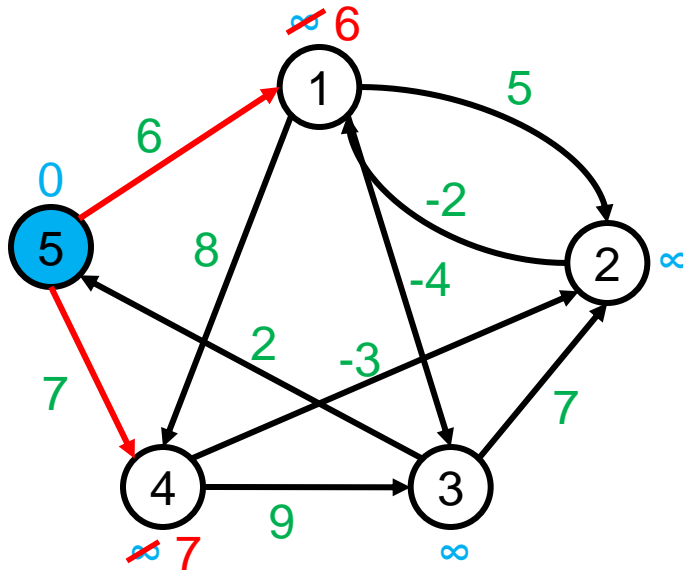
Exemplu Bellman-Ford (II)



$$d[1] = d[2] = d[3] = d[4] = d[5] = \infty$$
$$d[5] = 0$$

- BellmanFord(G,s)
 - Pentru fiecare v din V // init
 - $d[v] = \infty$;
 - $p[v] = \text{null}$;
 - $d[s] = 0$; // actualizare distanță până la s
 - Pentru i de la 1 la $|V| - 1$ // pt // fiecare pas de la s spre $V-s$
 - Pentru fiecare (u,v) din E // pt. // arcele ce pleacă de la nodurile // deja considerate
 - Dacă $d[v] > d[u] + w(u,v)$ atunci // se relaxează arcele // corespunzătoare
 - $d[v] = d[u] + w(u,v)$;
 - $p[v] = u$;
 - Pentru fiecare (u,v) din E
 - Dacă $d[v] > d[u] + w(u,v)$ atunci
 - Eroare ("ciclu negativ");

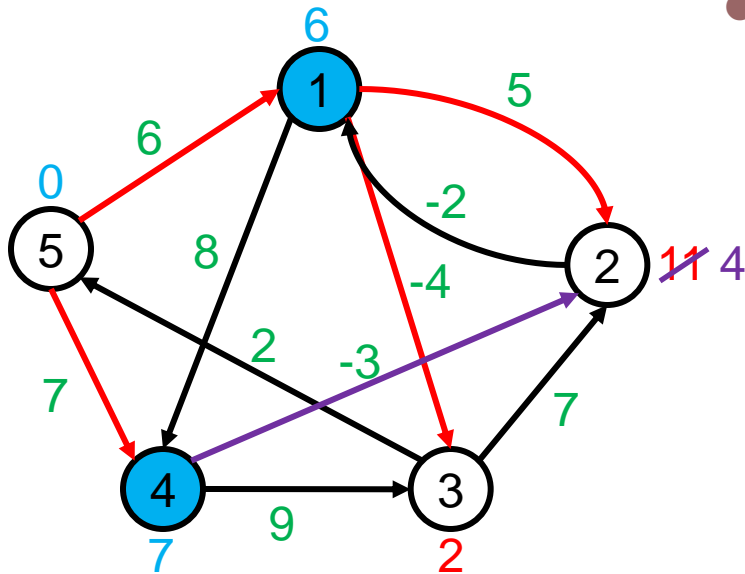
Exemplu Bellman-Ford (III)



Pas 1: relaxare (5,1) și (5,2)
 $d[1] = 6$, $p[1] = 5$
 $d[4] = 7$, $p[4] = 5$

- BellmanFord(G,s)
 - Pentru fiecare v din V // init
 - $d[v] = \infty$;
 - $p[v] = \text{null}$;
 - $d[s] = 0$; // actualizare distanță până la s
 - Pentru i de la 1 la $|V| - 1$ // pt // fiecare pas de la s spre $V-s$
 - Pentru fiecare (u,v) din E // pt. // arcele ce pleacă de la nodurile // deja considerate
 - Dacă $d[v] > d[u] + w(u,v)$ atunci // se relaxează arcele // corespunzătoare
 - $d[v] = d[u] + w(u,v)$;
 - $p[v] = u$;
 - Pentru fiecare (u,v) din E
 - Dacă $d[v] > d[u] + w(u,v)$ atunci
 - Eroare ("ciclu negativ");

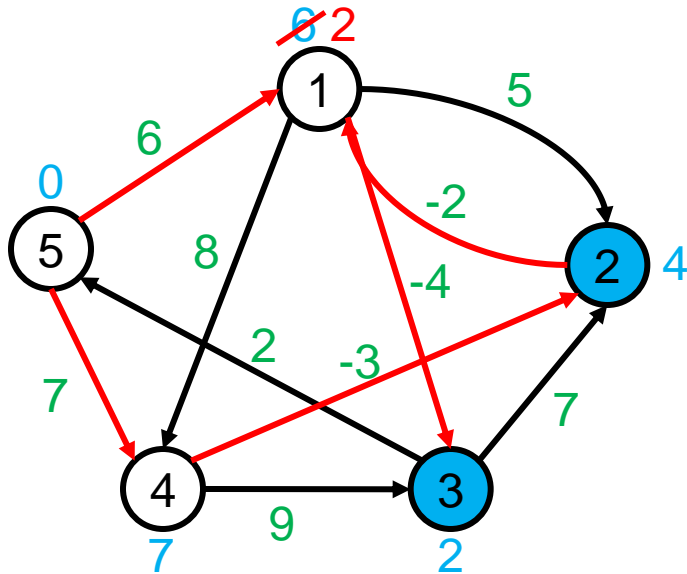
Exemplu Bellman-Ford (IV)



Pas 2: relaxare (1,2) și (1,3)
 $d[2] = 11$, $p[2] = 1$
 $d[3] = 2$, $p[3] = 1$
relaxare (4,2)
 $d[2] = 4$, $p[2] = 4$

- BellmanFord(G,s)
 - Pentru fiecare v din V // init
 - $d[v] = \infty$;
 - $p[v] = \text{null}$;
 - $d[s] = 0$; // actualizare distanță până la s
 - Pentru i de la 1 la $|V| - 1$ // pt // fiecare pas de la s spre $V-s$
 - Pentru fiecare (u,v) din E // pt. // arcele ce pleacă de la nodurile // deja considerate
 - Dacă $d[v] > d[u] + w(u,v)$ atunci // se relaxează arcele // corespunzătoare
 - $d[v] = d[u] + w(u,v)$;
 - $p[v] = u$;
 - Pentru fiecare (u,v) din E
 - Dacă $d[v] > d[u] + w(u,v)$ atunci
 - Eroare ("ciclu negativ");

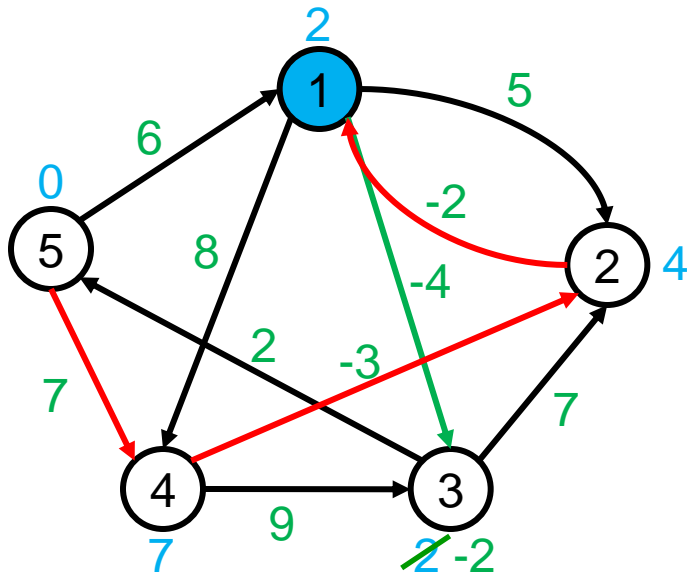
Exemplu Bellman-Ford (V)



Pas 3: relaxare (2,1)
 $d[1] = 2$, $p[1] = 2$

- BellmanFord(G,s)
 - Pentru fiecare v din V // init
 - $d[v] = \infty$;
 - $p[v] = \text{null}$;
 - $d[s] = 0$; // actualizare distanță până la s
 - Pentru i de la 1 la $|V| - 1$ // pt // fiecare pas de la s spre $V-s$
 - Pentru fiecare (u,v) din E // pt. // arcele ce pleacă de la nodurile // deja considerate
 - Dacă $d[v] > d[u] + w(u,v)$ atunci // se relaxează arcele // corespunzătoare
 - $d[v] = d[u] + w(u,v)$;
 - $p[v] = u$;
 - Pentru fiecare (u,v) din E
 - Dacă $d[v] > d[u] + w(u,v)$ atunci
 - Eroare ("ciclu negativ");

Exemplu Bellman-Ford (VI)



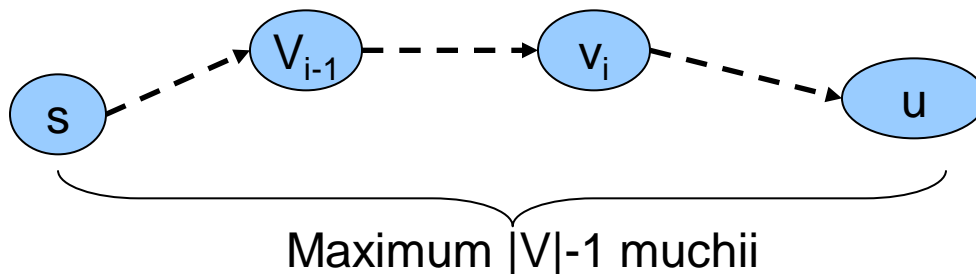
Pas 4: relaxare (1,3)
 $d[3] = -2$, $p[3] = 1$

- BellmanFord(G,s)
 - Pentru fiecare v din V // init
 - $d[v] = \infty$;
 - $p[v] = \text{null}$;
 - $d[s] = 0$; // actualizare distanță până la s
 - Pentru i de la 1 la $|V| - 1$ // pt // fiecare pas de la s spre $V-s$
 - Pentru fiecare (u,v) din E // pt. // arcele ce pleacă de la nodurile // deja considerate
 - Dacă $d[v] > d[u] + w(u,v)$ atunci // se relaxează arcele // corespunzătoare
 - $d[v] = d[u] + w(u,v)$;
 - $p[v] = u$;
 - Pentru fiecare (u,v) din E
 - Dacă $d[v] > d[u] + w(u,v)$ atunci
 - Eroare ("ciclu negativ");

Corectitudine Bellman-Ford (I)

- **Lemă 25.12:** $G = (V, E)$, $w : E \rightarrow \mathbb{R}$ funcție de cost asociată; dacă G **nu conține ciclu de cost negativ** atunci după $|V| - 1$ iterații ale relaxării fiecărei muchii avem $d[v] = \delta(s, v)$ pentru $\forall v \in R(s)$.
- **Dem prin inducție:**
 - Fie $s = v_0, v_1 \dots v_k = u$ un drum minim în graf cu $k \leq |V| - 1$.

La pasul i va fi relaxată muchia v_{i-1}, v_i



Corectitudine Bellman-Ford (II)

- **Demonstrăm că** in pasul i : $d[v_i] = \delta(s, v_i)$ și se menține până la sfârșit.
- P_0 : (inițializare) $\rightarrow d[s] = d[v_0] = 0 = \delta(s, s) = \delta(s, v_0)$ și conf. **Lema 25.5**, relația se menține până la sfârșit.
- $P_{i-1} \rightarrow P_i$:
 - P_{i-1} : $d[v_{i-1}] = \delta(s, v_{i-1})$,
 - În pasul i se relaxează muchia (v_{i-1}, v_i) , \Rightarrow conf. **Lema 25.7** $\Rightarrow d[v_i] = d[v_{i-1}] + (v_{i-1}, v_i) = \delta(s, v_{i-1}) + (v_{i-1}, v_i) = \delta(s, v_i)$. Conf. **Lema 25.5**, relația se menține până la sfârșit.
 - Cum $i \in (1, |V|-1) \rightarrow$ relația e adevărată pentru toate nodurile accesibile din $s \rightarrow d[v] = \delta(s, v), \forall v \in R(s)$.

Corectitudine Bellman-Ford (III)

- **Teorema.** $G = (V, E)$, $w : E \rightarrow \mathfrak{R}$ funcție de cost asociată. Algoritmul Bellman-Ford aplicat acestui graf plecând din sursa s nu returnează EROARE dacă G nu conține cicluri negative, iar la terminare $d[v] = \delta(s, v)$ pentru $\forall v \in V$. Dacă G conține cel puțin un ciclu negativ accesibil din s , atunci algoritmul întoarce EROARE.
- **Dem:** pe baza [Lemei 25.12](#).
 - Dacă \nexists ciclu negativ:
 - $d[v] = \delta(s, v) \forall v \in R(s)$
 - $d[v] = \delta(s, v) = \infty, \forall v \notin R(s)$ (inițializare)
 - $\rightarrow d[v] \leq d[u] + w(u, v) \rightarrow$ nu se întoarce eroare (conf. [Lema 25.3](#))
 - Dacă \exists ciclu negativ \rightarrow în cei $|V| - 1$ pași se scad costurile muchiilor, iar în final ciclul se menține \rightarrow Eroare

Optimizări Bellman-Ford

- **Observație!**
 - Dacă $d[v]$ **nu se modifică** la pasul i atunci **nu trebuie sa relaxăm** niciuna din muchiile care pleacă din v la pasul $i + 1$.
 - \Rightarrow păstrăm o coadă cu vârfurile modificate (o singură copie).

Bellman-Ford optimizat

- BellmanFordOpt(G,s)
 - Pentru fiecare v din V
 - $d[v] = \infty$;
 - $p[v] = \text{null}$;
 - $\text{marcat}[v] = \text{false}$; // marcăm nodurile pentru care am făcut relaxare
 - $Q = \emptyset$; // coadă cu priorități
 - $d[s] = 0$; $\text{marcat}[s] = \text{true}$; Introdu(Q,s);
 - Cât timp ($Q \neq \emptyset$)
 - $u = \text{ExtrageMin}(Q)$; $\text{marcat}[u] = \text{false}$; // extrag minimumul
 - Pentru fiecare (u,v) din E
 - Dacă $d[v] > d[u] + w(u,v)$ atunci // relaxez arcele ce pleacă din u
 - $d[v] = d[u] + w(u,v)$;
 - $p[v] = u$;
 - Dacă ($\text{marcat}[v] == \text{false}$) { $\text{marcat}[v] = \text{true}$; Introdu(Q,v);} }
- Observație: nu mai detectează cicluri negative!

Complexitate Bellman-Ford

- cazul defavorabil:

- Pentru i de la 1 la $|V| - 1$
- Pentru fiecare (u,v) din E
 - Dacă $d[v] > d[u] + w(u,v)$ atunci
 - $d[v] = d[u] + w(u,v)$;
 - $p[v] = u$;

V
*
 E
—
 $O(VE)$