ALGORITMICA GRAFURILOR
Săptămâna 4

C. Croitoru

croitoru@info.uaic.ro

FII

October 20, 2013

OUTLINE

1. **Probleme de drum în (di)grafuri**
   (ag 13-14 allinone.pdf pag. 73 $\longrightarrow \ldots$)

2. **Problemele pentru seminarul 4**

Drumuri de cost minim

P1 Date G digraf;$a : E(G) \to \mathbf{R}$; $s, t \in V(G)$,$s \neq t$.
Să se determine $D^*_{st} \in \mathcal{D}_{st}$, astfel încît
$a(D^*_{st}) = \min\{a(D_{st}) \mid D_{st} \in \mathcal{D}_{st}\}$.

P2 Date G digraf; $a : E(G) \to \mathbf{R}$; $s \in V(G)$.
Să se determine $D^*_{si} \in \mathcal{D}_{si}\ \forall i \in V(G)$, a.î.
$a(D^*_{si}) = \min\{a(D_{si}) \mid D_{si} \in \mathcal{D}_{si}\}$.

P3 Date G digraf; $a : E(G) \to \mathbf{R}$.
Să se determine $D^*_{ij} \in \mathcal{D}_{ij}\ \forall i, j \in V(G)$, a.î.
$a(D^*_{ij}) = \min\{a(D_{ij}) \mid D_{ij} \in \mathcal{D}_{ij}\}$.

Rezolvarea problemei P2

Teorema 1. Fie $G = (V, E)$ digraf, $V = \{1, \dots, n\}$, $s \in V$ și $a : E \to \mathbf{R}$, astfel încît

$$(I) \qquad \forall C \text{ circuit în } G, a(C) > 0.$$

Atunci $(u_1, \dots, u_n)$ **este o soluție a sistemului**

$$(*) \qquad \begin{cases} u_s = 0 \\ u_i = \min\limits_{j \neq i}(u_j + a_{ji}) \quad \forall i \neq s. \end{cases}$$

dacă și numai dacă
$\forall i \in V$, $\exists D^*_{si} \in \mathcal{D}_{si}$ astfel încît $a(D^*_{si}) = u_i$ și $a(D^*_{si}) = \min\{a(D) \mid D \in \mathcal{D}_{si}\}$.

Rezolvarea problemei P2 dacă G este digraf fără circuite

O **numerotare aciclică** a (vârfurilor) digrafului $G = (V, E)$ este un vector $ord[v] \quad v \in V$, (cu interpretarea $ord[v]$ = numărul de ordine al vârfului v) a. î.

$$\forall vw \in E \Rightarrow ord[v] < ord[w].$$

G este un digraf fără circuite dacă și numai dacă admite o numerotare aciclică .

Sortare topologică $\mathcal{O}(n + m)$.

Rezolvarea sistemului $(*)$ prin substituție.

Rezolvarea problemei P2 dacă $\forall ij \in E(G)$ avem $a_{ij} \geq 0$!!!

Algoritmul lui Dijkstra

1. $S \leftarrow \{s\}$; $u_s \leftarrow 0$; $\hat{\imath}nainte[s] \leftarrow 0$;
   for $i \in V \setminus \{s\}$ **do**
   $\{ \quad u_i \leftarrow a_{si}$; $\hat{\imath}nainte[i] \leftarrow s$ $\}$
   // după aceste inițializări (D) are loc
2. **while** $S \neq V$ **do**
   $\{$
   determină $j^* \in V \setminus S$: $u_{j^*} = \min\{u_j \mid j \in V \setminus S\}$;
   $S :\leftarrow S \cup \{j^*\}$;
   for $j \in V \setminus S$ **do**
   if $u_j > u_{j^*} + a_{j^*j}$ **then**
   $\{ \quad u_j \leftarrow u_{j^*} + a_{j^*j}$; $\hat{\imath}nainte[j] \leftarrow j^*$ $\}$
   $\}$

Complexitatea timp a algoritmului, în descrierea dată este $O(n^2)$.

Algoritmul lui Dijkstra

Este posibilă organizarea unor cozi cu prioritate (de exemplu heap-urile) pentru obținerea unui algoritm cu complexitatea $O(m \log n)$ (unde $m = |E|$) *Johnson ,1977).*

Cea mai bună implementare se obține utilizând **heap-uri Fibonacci**, ceea ce conduce la o complexitate timp de $O(m + n \log n)$ (*Fredman și Tarjan, 1984*).

Drumuri de cost minim

Rezolvarea problemei P2 în cazul general.

Algoritmul lui *Bellman, Ford, Moore* ($\sim$ 1960)

1. $u_s^1 \leftarrow 0$; **for** $i \in V \setminus \{s\}$ **do** $u_i^1 \leftarrow a_{si}$;
   // evident (BM) are loc
2. **for** $m := 1$ **to** $n-2$ **do**
   for $i := 1$ **to** n **do**
   $u_i^{m+1} \leftarrow \min\left(u_i^m, \min_{j \neq i}(u_j^m + a_{ji})\right)$

Complexitatea $O(n^3)$, dacă determinarea minimului din pasul 2 necesită $O(n)$ operații.

Testarea în $O(n^3)$ a existenței unui circuit C de cost negativ în digraful G !

Problemele pentru seminarul 4

1. **Problema 1, Setul 3”**
2. **Problema 3, Setul 6**
3. **Problema 2, Setul 20**
4. **3-4 probleme din lista următoare :)**

Probleme pentru seminarul 4

1

Să se construiască o funcție care să recunoască un turneu. La intrare aceasta va primi un digraf $G = (\{1, ..., n\}, E)$ reprezentat cu ajutorul listelor de adiacență și va returna *true* sau *false*. Complexitatea timp?.

2

Să se construiască o funcție care primind la intrare un digraf $G = (\{1, ..., n\}, E)$ reprezentat cu ajutorul listelor de adiacență să returneze inversul lui G reprezentat cu ajutorul listelor de adiacență. Complexitatea timp trebuie să fie $\mathcal{O}(n + |E|)$.

3

Se consideră un graf $G = (\{1, ..., n\}, E)$ reprezentat cu ajutorul matricii de adiacență. Mulțimea de $n - 1$ muchii A este astfel ca $T = (V, A)$ este arbore parțial al lui G. Construiți un algoritm care să listeze circuitele care se formează prin adăugarea muchiilor din $E - A$ la T. Reprezentarea lui T trebuie să permită depistarea fiecărui astfel de circuit în timpul $\mathcal{O}(n)$.

Probleme pentru seminarul 4

4

Să se construiască o funcție care să determine gradul maxim al unui vârf al unui graf. La intrare aceasta va primi un graf $G = (\{1, ..., n\}, E)$ reprezentat cu ajutorul listelor de adiacență și va returna $\Delta(G)$. Stabiliți complexitatea timp a algoritmului folosit.

5

Construiți o funcție care primind la intrare graful $G = (V, E)$ reprezentat cu ajutorul listelor de adiacență și k, un număr întreg pozitiv, returnează graful $G^{(k)}$ cu aceeași mulțime de virfuri ca și G , în care două virfuri distincte sunt adiacente dacă și numai dacă în graful inițial sunt conectate printr-un drum de lungime cel mult k. Care este complexitatea timp?

6

Graful conex $G = (V, E)$ cu n vârfuri și m muchii, este reprezentat cu ajutorul listelor de adiacență. Dați un algoritm care să construiască în timpul $O(n + m)$ listele de adiacență ale unui arbore parțial al lui G.

7

Fie $G = (V, E)$ un graf cu ordinul $|V| \geq 2$ și $T = (V, E_T)$ un arbore parțial al lui G, dat de tabloul $(p[v])_{v \in V}$, unde $p[v]$ este părintele lui v în T: vârful dinaintea lui v de pe drumul unic de la o rădăcină fixată r, la v, în T $(p[r] = r)$. Dați un algoritm care să determine, în timpul $\mathcal{O}(|V|)$, un vârf v_0 pendant (frunză) în T și apoi demonstrați că $G - v_0$ este conex.

8

Digraful $G = (V, E)$ este dat prin listele de adiacență. Să se decidă în $O(|V| + |E|)$ dacă se pot ordona vârfurile sale: $v_{i_1}, \ldots, v_{i_{|V|}}$, astfel încât dacă v_{i_j} apare în lista de adiacență a lui v_{i_k} atunci $k < j$.

9

Fie $T = (\{1, \ldots, n\}, E_T)$ un arbore $(n \geq 2)$, dat de tabloul $(p[v])_{v \in V}$, unde $p[v]$ este părintele lui v în T: vârful dinaintea lui v de pe drumul unic de la o rădăcină fixată r, la v, în T $(p[r] = r)$. Descrieți un algoritm care să construiască , în timpul $\mathcal{O}(n)$, listele de adiacență ale lui T.

Probleme pentru seminarul 4

10

Se consideră un graf $G = (V, E)$ ($V = \{1, \ldots, n\}$), izomorf cu graful circuit (cu cel puțin 3 vârfuri), $G \cong C_n$. Fiecare muchie $e \in E$ are asociat un cost real $c(e)$. Aceste informații sunt disponibile în tablourile *dreapta* și *cost* de dimensiune n cu semnificația: $dreapta[v]$ = vecinul din dreapta al lui v, iar costul muchiei $\{v, dreapta(v)\}$ este $cost[v]$. Descrieți un algoritm cât mai eficient pentru aflarea unui arbore parțial al lui G de cost minim.

11

Se consideră un graf $G = (V, E)$ ($V = \{1, \ldots, n\}$), reprezentat cu ajutorul listelor de adiacență. Se știe că graful are gradul minim $\delta(G)$ mărginit de o constantă $c \in \mathbf{N}$. Descrieți un algoritm cu complexitatea timp $O(n)$ pentru determinarea lui $\delta(G)$ și a unui vârf $v_0 \in V$ cu gradul în G egal cu $\delta(G)$.