LUCRAREA DE LABORATOR Nr. 4

TEMA: ALGORITMI DE DETERMINARE A DRUMULUI MINIM

1. SCOPUL LUCRĂRII:

- Studierea algoritmilor de determinare a drumurilor minime într-un graf.
- Elaborarea programelor de determinare a drumului minim într-un graf ponderat.

2. SARCINA DE BAZĂ

1. Elaboraţi procedura introducerii unui graf ponderat;
2. Elaboraţi procedurile determinării drumului minim conform algoritmului Ford;
3. Realizaţi un program cu următoarele funcţii:
   - introducerea grafului ponderat cu posibilităţi de analiză sintactică şi semantică şi de corectare a informaţiei;
   - determinarea drumului minim prin algoritmul Ford;
   - extragerea informaţiei (valoarea drumului minim şi succesiunea vârfurilor care formează acest drum).

- Lucrarea va fi prezentată la lecția de laborator și plasată pe Teams.
- Lucrarea va fi denumită după formatul NumePrenumeGrupa.c
- Prezentarea cu întârziere se penalizează

3. NOTE DE CURS

Noţiune de drum minim

Pentru un graf orientat $G = (X,U)$ se va numi *drum* un şir de vârfuri $D = (x_0, x_1,..., x_r)$ cu proprietatea că $(x_0, x_1), (x_1, x_2),..., (x_{r-1}, x_r)$ aparţin lui U, deci sunt arce ale grafului şi extremitatea finală a arcului precedent coincide cu extremitatea iniţială a arcului următor.

Vârfurile x_0 şi x_r se numesc extremităţile drumului D. Lungimea unui drum este dată de numărul de arce pe care le conţine. Dacă vârfurile $x_0, x_1,..., x_r$ sunt distincte două câte două drumul D este elementar.

Adeseori, fiecărui arc (muchii) i se pune în corespondenţă un număr real care se numeşte *ponderea* (lungimea) arcului. Lungimea arcului (x_i, x_j) se va nota $w(i,j)$, iar în cazul în care un arc este lipsă ponderea lui va fi considerată foarte mare (pentru calculator cel mai mare număr pozitiv posibil). În cazul grafurilor cu arce ponderate (grafuri ponderate) se va considera lungime a unui drum suma ponderilor arcelor care formează acest drum. Drumul care uneşte două vârfuri concrete şi are lungimea cea mai mică se va numi *drum minim* iar lungimea drumului minim vom numi *distanţă*. Vom nota distanţa dintre x şi t prin $d(x, t)$, evident, $d(x,x)=0$.

Algoritmul lui Ford pentru detrminarea drumului minim

Permite determinarea drumului minim care începe cu un vârf iniţial x_i până la oricare vârf al grafului G. Dacă prin L_{ij} se va nota ponderea arcului (x_i, x_j) atunci algoritmul conţine următorii paşi:

1. Fiecărui vârf x_j al grafului G se va ataşa un număr foarte mare $H_j(\infty)$. Vârfului iniţial i se va ataşa $H_o = 0$;

2. Se vor calcula diferenţele $H_j - H_i$ pentru fiecare arc (x_i, x_j). Sunt posibile trei cazuri:

   a) $H_j - H_i < L_{ij}$,

   b) $H_j - H_i = L_{ij}$,

c) $H_j - H_i > L_{ij}$.

Cazul "*c*" permite micşorarea distanţei dintre vârful iniţial şi x_j din care cauză se va realiza $H_j = H_i + L_{ij}$.

Pasul 2 se va repeta atâta timp cât vor mai exista arce pentru care are loc inegalitatea "c". La terminare, etichetele H_i vor defini distanţa de la vârful iniţial până la vârful dat x_i.

3. Acest pas presupune stabilirea secvenţei de vârfuri care va forma drumul minim. Se va pleca de la vârful final x_j spre cel iniţial. Predecesorul lui x_j va fi considerat vârful x_i pentru care va avea loc $H_j - H_i = L_{ij}$. Dacă vor exista câteva arce pentru care are loc această relaţie se va alege la opţiune.

Algoritmul Bellman - Calaba

Permite determinarea drumului minim dintre oricare vârf al grafului până la un vârf, numit vârf final.

Etapa iniţială presupune ataşarea grafului dat *G* a unei matrice ponderate de adiacenţă, care se va forma în conformitate cu următoarele:

1. $M(i,j) = L_{ij}$, dacă există arcul (x_i, x_j) de pondere L_{ij};
2. $M(i,j) = \infty$, unde ∞ este un număr foarte mare (de tip întreg maximal pentru calculatorul dat), dacă arcul (x_i, x_j) este lipsă;
3. $M(i,j) = 0$, dacă $i = j$.

La etapa a doua se va elabora un vector V_0 în felul următor:

1. $V_{0(i)} = L_{in}$, dacă există arcul (x_i, x_n), unde x_n este vârful final pentru care se caută drumul minim, L_{in} este ponderea acestui arc;
2. $V_{0(i)} = \infty$, dacă arcul (x_i, x_n) este lipsă;
3. $V_{0(i)} = 0$, dacă $i = j$.

Algoritmul constă în calcularea iterativă a vectorului *V* în conformitate cu următorul procedeu:

1. $V_{k(i)} = min\{V_{k-1}; L_{ij}+V_{k-1(j)}\}$, unde $i = 1, 2,..., n - 1, j = 1, 2,..., n$; $i<>j$;
2. $V_{k(n)} = 0$.

Când se va ajunge la $V_k = V_{k-1}$ - STOP.

Componenta cu numărul *i* a vectorului V_k cu valoarea diferită de zero ne va da valoarea minimă a drumului care leagă vârful *i* cu vârful *n*.

SARCINA DE BAZĂ

4. Elaboraţi procedura introducerii unui graf ponderat;
5. Elaboraţi procedurile determinării drumului minim conform algoritmului Ford;
6. Realizaţi un program cu următoarele funcţii:

   - introducerea grafului ponderat cu posibilităţi de analiză sintactică şi semantică şi de corectare a informaţiei;
   - determinarea drumului minim prin algoritmul Ford;
   - extragerea informaţiei (valoarea drumului minim şi succesiunea vârfurilor care formează acest drum).

4. ÎNTREBĂRI DE CONTROL

1. Ce se numeşte graf ponderat?
2. Definiţi noţiunea de distanţă.
3. Descrieţi etapele principale ale algoritmului Ford.

4. Care sunt momentele principale în algoritmul Bellman-Calaba?
5. Prin ce se deosebeşte algoritmul Ford de algoritmul Bellman-Calaba?
6. Cum se va stabili succesiunea vârfurilor care formează drumul minim?