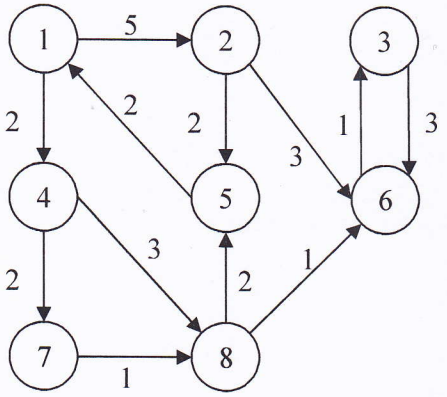
PROIECTAREA ALGORITMILOR
Examen – 03.06.2011
Timp de lucru: 1h20

SUBIECTUL 1 (11p = 4 x 2p + 1 x 3p) – 25 min



Se da graful din imaginea alaturata. Pentru punctul 1, graful se considera neorientat.

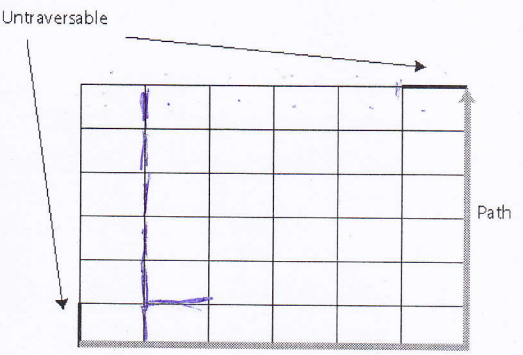
1. Sa se aplice algoritmul lui Prim, pornind din nodul 1, specificand la fiecare pas nodul extras din coada si nodurile aflate in coada de prioritati, impreuna cu prioritatea asignata fiecaruia (vectorul d);
2. Aplicati primii 3 pasi ai algoritmului Floyd-Warshall, evidentiand valorile matricilor $D^{(0)}$, $D^{(1)}$, $D^{(2)}$, $D^{(3)}$. La fiecare pas, este suficient sa spuneti care elemente din matricea de la pasul anterior se modifica.
3. Sa determine tipul muchiilor grafului folosind o parcurgere in adancime (ignorand costurile asociate muchiilor), pornind din nodul 1 (specificand si timpii de descoperire si de finalizare pentru fiecare nod);
4. Sa se determine componentele tari conexe ale grafului, pornind de la parcurgerea in adancime anterioara si aplicand a doua etapa a algoritmului lui Kosaraju;
5. Sa se aplice algoritmul A*, considerand starea initiala nodul s = 5 si starea finala nodul t = 6. Euristica folosita este h(v) = {min(cost(v, v')) | (v, v') este muchie in graf} pentru v != t si h(t) = 0. Sa se calculeze valorile euristicii h. Sa se evidentieze, la fiecare pas, multimile OPEN si CLOSED, precum si modificarile efectuate asupra f si g.

SUBIECTUL 2 (15p = 5 x 3p) – 25/30 min

1. Comparatie intre tehnicile de programare lacoma si programare dinamica.
2. Pornind de la motivul pentru care algoritmul AC3 este mai bun decat AC1, comparati cei doi algoritmi prezentand avantajele si dezavantajele fiecaruia (unde se folosesc, complexitati, etc.).
3. Din ce motiv algoritmul lui Dijkstra nu poate fi folosit pentru grafuri ce contin muchii de cost negativ ce pot fi atinse din sursa? Dati un exemplu simplu de astfel de graf pentru care algoritmul lui Dijkstra ar intoarce rezultatul gresit.
4. Ce se intampla cu excedentul de flux existent intr-o retea de transport dupa ce algoritmul de pompare preflux a atins fluxul maxim (adica nu mai exista nici o cale reziduala catre drena)?
5. Explicati principiul taieturilor alfa-beta si care este imbunatatirea adusa comparand (analiza a complexitatii) algoritmul minimax cu alfa-beta cu algoritmul minimax simplu.

SUBIECTUL 3 (14p) – 25/30 min

In New Bucharest, strazile sunt construite sub forma de grid de dimensiune ***lung_x*** si ***lung_y***. Din diverse motive, unele din aceste strazi nu sunt practicabile – puteti considera ca exista un vector ***inutilizabil*** care contine aceste strazi. Pornind din coordonata ***(0, 0)***, scopul vostru este sa determinati numarul de cai (drumuri) distincte existente pana la destinatia ***(lung_x, lung_y)***. Mai mult decat atat, fiecare cale trebuie sa aiba exact ***lung_x*** + ***lung_y*** segmente (strazi).



Pentru subiectul 3, cerintele sunt urmatoarele:
a. Explicati mai intai care este ideea voastra de rezolvare (in cuvinte, incercati sa oferiti cat mai multe detalii);
b. Spuneti ce structuri de date, tehnici de programare si/sau algoritmi clasici (prezentati la curs) veti folosi;
c. Schitati pseudocodul, fara a intra in detalii (inutile).
Atentie: Se doreste o complexitate cat mai buna! Determinati complexitatea solutiei propuse!