Vitoga George Patrick

2.5 Drumuri minime in graf

1. Solutiile alese:

Vom trata "problema gasirii tuturor distantelor dintre un nod si toate celelalte" cu ajutorul algoritmilor: Dijkstra, BellmanFord si Dijkstra-adaptat .

2. Aplicatii practice:

Aceasta problema este des intalnita in multe aplicatii practice :

- **reteaua de telefonie**: liniile de telefonie au o anumita latime de banda, iar noi vrem sa directionam un apel prin linia cu cea mai mare latime de banda

- **curierat** : stim bugetul alocat pentru o cursa si costul pt a efectua un trasport intre doua locatii si vrem sa aflam in cate orase ne incadram sa livram de la sediu

- aplicatii simplificate ale problemei : aflarea **drumului minim** de la inceput la destinatie (aviatie ex care e cea mai rapida sau ieftina metoda sa zburam din aeroport A in aeroport B)

3. Descriere algoritmi:

Dijkstra si Dijkstra-adaptat :

- Folosit pentru aflarea distantei pornind dintr-un anumit nod la toate celelalte, vulnerabil in grafurile cu cicluri negative, mai multe implementari (ex: relaxare si heap, heap Fibonacci), este greedy, complexitate **O(**noduri * log(noduri)**)**

BellmanFord :

- Folosit pentru aflarea distantei pornind dintr-un anumit nod la toate celelalte, adaptat pentru grafuri cu drumuri negative, mai simplu decat Dijkstra dar cu complexitate mai mare **O(**noduri * muchii**)**

4. Verificare si testare:

Dupa implementarea algoritmilor (Dijkstra, BellmanFord si Dijkstra-adaptat), voi folosi seturi mici de date, usor de calculat manual pentru a vedea comportamentul solutiilor implementate. Daca toate testele se sfarsesc cu bine, o sa iau seturi de date facute de Standford (http://snap.stanford.edu/data/index.html) sau cu ajutorul unui generator si calculator de graph uri (https://graphonline.ru/en/). Daca solutia va continua sa functioneze, o sa fac un program in python care genereaza graph-uri prefabricate (intai o sa generez eu un traseu "S" iar dupa, cand generez restul graph-ului, o sa am grija ca orice alt traseu, generat random din conexiuni, va fi mai mare decat traseul "S". Astfel o sa imi asigur validitatea solutiei "S", solutie pe care va tribui sa o gaseasca algoritmul meu, daca incerc sa calculez distanta minima dintre 2 noduri. Ca sa generalizez, o sa generez n solutii de tip "S", si voi adauga muchii aditionale, cu dinstante vulnerabile, asigurandu-mi iar validitatea solutiei).

5. Referinte:

GeekForGeeks:

https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/

StackOverFlow:

https://stackoverflow.com/questions/42178106/find-distance-from-node-to-all-other-nodes-in-a-weighted-graph

Standfort:

Graph Algorithms - Virginia Williams http://theory.stanford.edu/~virgi/cs267/index.html

Carti:

"Cracking the coding inteview" - Gayle Laakmann McDowell

"Introducere in Analiza Algoritmilor" - Cristian Giumale