Grade – solutie

Problema construirii unui graf dintr-o lista de grade este cunoscuta iar algoritmul ar consta in urmatorii pasi :

1- Se sorteaza gradele in ordine descrescatoare ; fie d1,d2,d3,… lista sortata
2- se considera ca nodul 1 are gradul d1 si este adiacent cu nodurile 2,3,…,d1+1; in consecinta se scade 1 din valorile d2,d3,…,d_{d1+1} iar d1 este eliminat din lista.
3- se repeta pasii 1 si 2 pana cand toate gradele devin 0.

Acest algoritm insa nu garanteaza obtinerea unui graf conex , (mai ales pentru liste de noduri care caracterizeaza grafuri cu numar mic de muchii); de exemplu, pentru lista de grade 3,2,2,1,1,1 aplicand algoritmul de mai sus se obtine graful cu muchiile :
1-2, 1-3, 1-4, 2-3, 5-6.
In plus, complexitatea este $n^2 \log n$ ceea ce duce la depasirea timpului de rulare !

Pentru a garanta conexitatea, algorimul de mai sus se modifica in modul urmator :

1- Se sorteaza gradele in ordine descrescatoare ; fie d1,d2,d3,… lista sortata
2- se considera ca nodul 1 are gradul d1 si este adiacent cu nodurile 2,3,…, dar in cazul in care exista mai multe valori egale cu d1, se considera ca 1 este adiacent cu ultimul (ca numar de ordine) dintre nodurile care au gradul d1 ; in consecinta se scad 1 din valorile d2,d3,…,
3- se repeta pasul 2 pana cand toate gradele devin 0.

Aplicand acest algoritm pentru exemplul anterior, se vor obtine muchiile :
1-2, 1-3, 1-6, 2-5, 3-4 care sunt ale unui graf conex !
Observam ca procedand astfel, nu mai este necesara re-sortarea: transformarile succesive ale listei de grade sunt :
3 2 2 1 1 1
0 1 1 1 1 0 (deci nodul 1 va fi adiacent cu 2, 3 si 6)
0 0 1 1 0 0 (nodul 2 adiacent cu 5)
0 0 0 0 0 0 (nodul 3 adiacent cu 4)

Totusi ramane problema incadrarii in timp, deoarece complexitatea ramane $n^2 \log n$;
Pentru inceput, sa observam ca daca n=1000, numarul de muchii ar putea fi de ordinul n^2 ceea ce ar face ca orice algoritm sa nu se incadreze in timp. Asa ca este de asteptat ca numarul de muchii in testele de intrare sa fie de ordinul n si nu n^2. (Acest lucru s-a impus si pentru a da teste la care algoritmul clasic sa construiasca grafuri neconexe !).
Pentru a reduce complexitatea la nlogn, trebuie evitata parcurgerea repetata a liste de grade pentru localizarea elementelor din stanga palierelor ; aceasta se face prin mentinerea unei liste de pointeri catre pozitiile de inceput si sfarsit ale fiecarui palier.