**Tema 2 - Desenare graf aleatoriu**

Sa se deseneze un graf plecand de la numarul de noduri, *n*, numarul de drumuri intre nodul initial si nodul final, *nPaths*, si numarul de cicluri din graf, *nCycles* .

1. Se construiesc drumuri elementare aleatoare intre primul si ultimul nod - 2p

s = 0;  
t = n −1;  
/\* only source is initially part of the path \*/  
for each node j other than s do  
 pathnode[j] = false;  
end for;  
pathnode[s] = true;  
/\* build the random path \*/  
u = s;  
for j = 1 to n-1 do  
/\* choose a random index k of the next node to be added to the path \*/  
 k = random(0, n–j-1);  
 l = 0;  
/\* find node v as the k-th node out of the not before chosen nodes \*/  
 for each node v do  
 if pathnode[v] then  
 continue;  
 end if;  
 if l = k then  
 break;  
 end if;  
 l = l + 1;  
 end for;  
/\* add arc (u,v) to the network \*/  
 ma[u][v] = 1;  
/\* mark node v as being part of path \*/  
 pathnode[v] = true;  
/\* if the last node v added to the path is sink, then path is completed \*/  
 if v = t then  
 break;  
 end if;  
/\* node u becomes v in order to prepare the adding of another node to the path \*/  
 u = v;  
end for;

1. Se construiesc cicluri aleatoare in graf - 2p

/\* choose a random node u0 \*/  
u0 = random(0, n-1);  
/\* only node u0 is initially part of the cycle \*/  
for each node j other than u0 do  
 cyclenode[j] = false;  
end for;  
/\* build the random cycle \*/  
u = u0;  
for j = 0 to n-1 do  
/\* choose a random index k of the next node to be added to the cycle \*/  
 k = random(0, n–j-1);  
 l = 0;  
/\* find node v as the k-th node out of the not before chosen nodes \*/  
 for each node v do  
 if cyclenode[v] then  
 continue;  
 end if;  
 if l = k then  
 break;  
 end if;  
 l = l + 1;  
 end for;  
/\* if v is u then regenerate v. This can only happen when u = u0 \*/  
 if u = v then  
 j = j – 1;  
 else  
/\* add arc (u,v) to the network \*/  
 ma[u][v] = 1;  
/\* mark node v as being part of cycle \*/  
 cyclenode[v] = true;  
 end if;  
/\* if v is the first chosen node u0, then cycle is completed \*/  
 if v = u0 then  
 break;  
 end if;  
/\* node u becomes v in order to prepare the adding of another node to the cycle \*/  
 u = v;  
end for;

1. Se genereaza nPaths drumuri si nCycles cicluri - 1p

4. Se genereaza lista de adiacenta - 1p

5. Se salveaza in fisier astfel: pe prima linie numarul de noduri, incepand cu cea de-a doua linie se salveaza lista de adiacenta -1p

6. Se deseneaza graful: pozitia nodurilor se alege fie aleatoriu (verificand ca nodurile sa nu se suprapuna)(2p) , fie nodurile se pozitioneaza pe un cerc, se traseaza mai apoi arcele intre noduri (3p) .

**Observații:**

1. Limbajul de programare este la alegerea voastra.
2. Temele copiate sau neprezentate se vor nota cu nota 1
3. Cerințele 1 - 2 sunt pentru nota 5
4. Cerințele 3 - 6 nu vor fi evaluate daca nu functioneaza cerintele 1-2;
5. Se poate primi 1p suplimentar daca se gaseste o metoda mai rapida de a genera drumuri si cicluri elementare (Argumentati)
6. Punctajul maxim care se poate obtine este 12p.
7. Timp de lucru - 2 saptamani