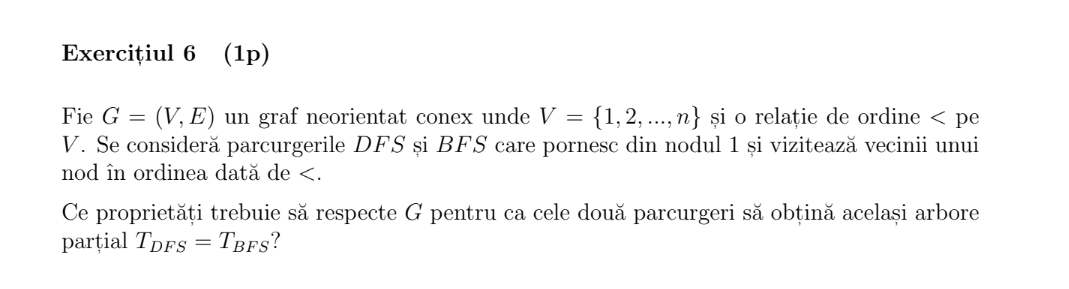
**Tema seminar 3**



Pentru ca graful sa aiba aceasi parcurgere in latime ca in adancime trebuie ca acesta sa fie un arbore (conex si fara cicluri). Daca G este arbore atunci arborele partial generat atat de parcurgerea BFS cat si DFS va fi egal cu el (si implicit egale si intre ele).

O demonstratie intuitiva ar fi:

Presupunem ca G nu e arbore => G are macar un ciclu

Plecand din nodul 1 la parcurgerea DFS cand dam de un ciclu el va fi parcurs circular. Sa zicem ca am ajuns in nodul x placand din 1 care face parte dintr-un ciclu. La BFS se vor pune cele doua noduri vecine cu nodul x in arborele parital (asemantor cu o parcurgere bifurcata). Astfel, vor rezulta doi arbori partiali diferiti pentru cele doua parcurgeri => G trebuie sa fie arbore.

Exemplu:

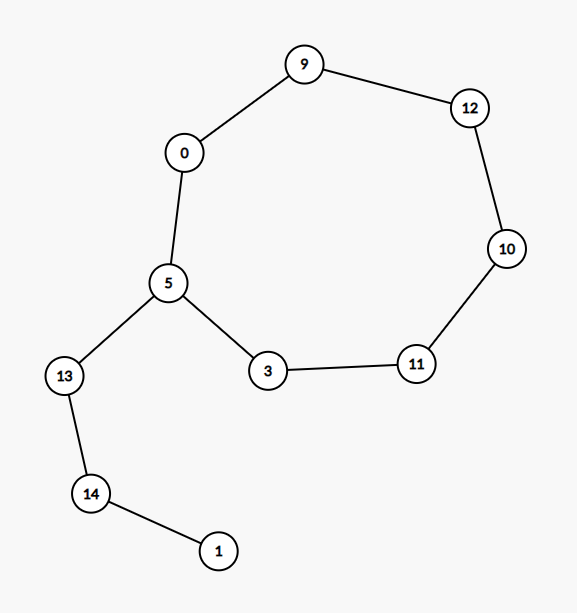
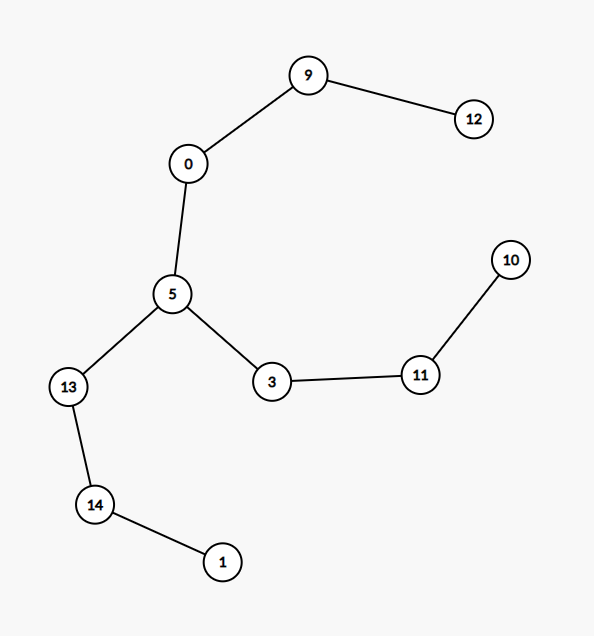
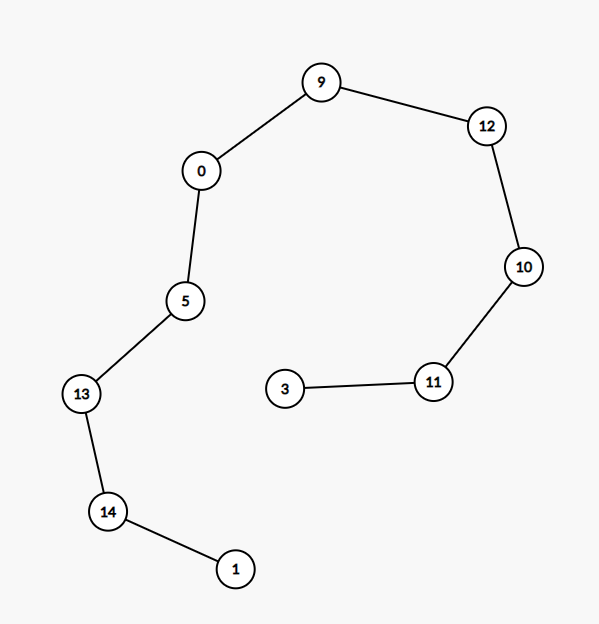
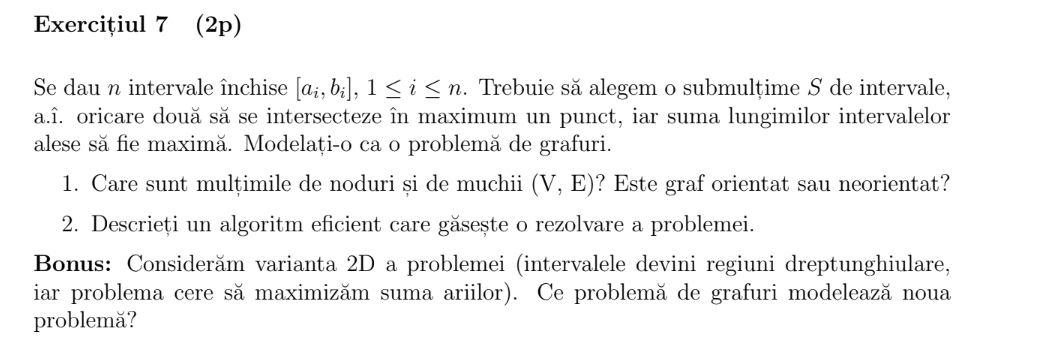
Pentru graful din figura 1 BFS va genera arborele partial din figura 2, iar DFS va genera arborele partial din figura 3.

Figura 1

Figura 2

Figura 3



1. Multimea de noduri este formata din toate capetele intervalelor (vom lua numai nodurile distincte) la care se mai adauga doua noduri auxiliare S si P.

Multimea de muchii este formata din:

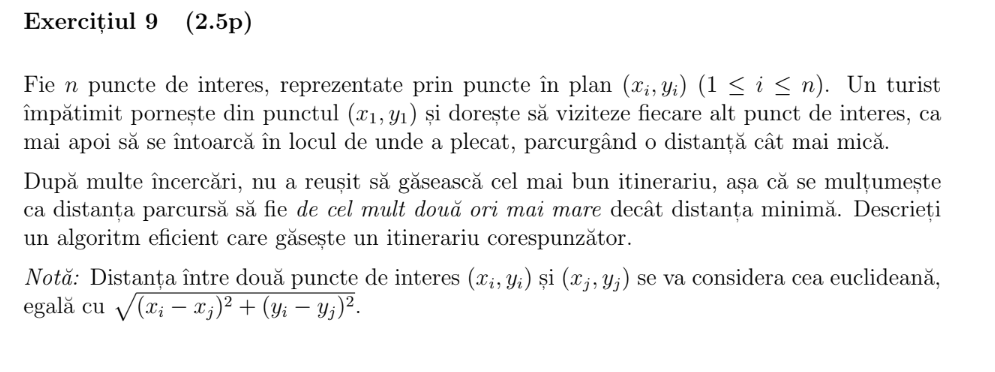
* muchiile de la ai la bi, 1 <= i <= n cu costul egal cu lungimea intervalului (b-a+1)
* muchiile de la S la toate nodurile ai cu costul 0
* muchiile de la toate nodurile bi la P cu costul 0
* muchiile de la bi la aj cu costul 0 care respecta proprietatea din cerinat (bi <= aj, 1 <= i, j <= n, i != j)

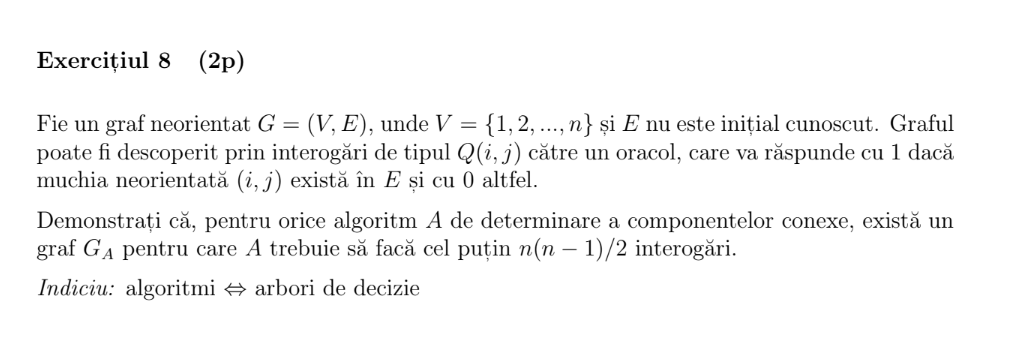
Graful este orientat.

1. Algoritmul eficient de rezolvare consta in gasirea drumului de cost maxim de la S la P.

Bonus:

Multumea nodurilor va fi formata din toate punctele (a, b) date si nodurile auxiliare S si P. Graful va fi tot orientat. Vom avea muchie de la fiecare nod la toate celelalte noduri aflate la dreapta jos fata de el cu costul egal cu aria de formata de un dreptunghi care sa contina ca colturi cele doua puncte din noduri. Vom avea muchie si de la S la toate nodurile care sunt punctul stanga sus dintr-un dreptunghi si de la toate nodurile care sunt punctul dreapta jos dintr-un dreptunghi la P. Calculez apoi drumul de cost maxim de la S la P.

Pentru a determina componentele conexe trebuie mai intai sa determinam multimea de muchii E. Cand interogam oracolul aflam informatie atat despre muchia de la x la y cat si de la y la x (graf neorientat), dar nu ne da informatie despre restul muchiilor. Din cauza aceasta, va trebui sa luam toate perechile de doua noduri (in care nu conteaza ordinea ca graful e neorientat) si sa intrebam oracolul daca avem muchie intre ele. Deci ca sa aflam toate muchiile sunt necesare cel putin C(n, 2) = = interogari.

Construim un graf complet G in care nodurile o sa fie cele n puncte de interes iar muchiile o sa aiba costul egal cu distanta de la fiecare punct de interes la toate celelalte puncte de interes.

Cel mai scurt traseu foloseste arbori partial de cost minim printr-o parcurgere DFS din nodul de start. Apoi inmultim cu 2 costul total al arborelui sau adaugam de doua ori fiecare muchie (o data cand parcurgem muchia si o data cand ne intorcem din recursie) ca sa aflam distanta minima pe care o are de parcurs.

Ramane sa demonstram ca traseul care parcurge eulerian APMul este drumul de cost minim. Presupunem prin reducere la absurd ca costul drumului minim este mai mic ca suma costurilor din APM. Dar cum APMul are drumul de cost minim si ambele sunt conexe => contadictie

Parcurgerea euleriana trece prin fiecare muchie de doua ori, deci costul total este de 2 \* suma costurilor din APM, care este mai mica sau egala cu 2 \* suma de pe drumul minim. Putem calcula APM-ul folosind algoritmul lui Prim.