**Tema - Divide et Impera**

1. Să se implementeze algoritmul căutării binare. (1p)

2. Să se implementeze algoritmul de sortare rapidă Quicksort (1p)

3. Să se implementeze algoritmul de sortare rapidă prin interclasare - Mergesort (1p)

4. Să de determine al k-lea element în ordine crescătoare dintr-un vector v cu n elemente, fără a sorta complet vectorul. (2p)

5. **Turnurile din Hanoi (1p)**

Problema turnurilor din Hanoi: Se dau trei tije numerotate cu 1, 2 şi 3. Pe tija 1 se găsesc n discuri de diametre diferite, aşezate în ordine descrescătoare a diametrelor privite de jos în sus. Se cere să se mute toate discurile de pe tija 1 pe tija k (k=2 sau 3) respectând următoarele reguli:

* La fiecare pas se mută un singur disc de pe o tijă pe alta
* Nu este permisă aşezarea unui disc de diametru mai mare peste un disc de diametru mai mic

Să se afişeze ordinea de efectuare a mutărilor.

Exemple (mutare de pe 1 pe 2)

* pentru n=1: se mută discul de pe 1 pe 2: (1,2)
* pentru n=2: se mută primul disc de pe 1 pe 3, apoi al doilea de pe 1 pe 2 şi din nou primul de pe 3 pe 2: (1,3)(1,2)(3,2)
* pentru n=3: (1,2)(1,3)(2,3)(1,2)(3,1)(3,2)(1,2)

6. **Determinarea maximului dintr-un vector „convex” (2p)**

Numim un vector convex, dacă elementele sale sunt in ordine este strict crescătoare până la o anumită pozişie după care sunt strict descrescătoare.

Exemplu: v ={1,2,3,4,5,6,4,3,2}

Să se determine maximul din vector cu ajutorul unui algoritm de tip „divide et impera”.

7. Al n-lea termen din şirul lui Fibonacci. (3p)

Definiţie recursivă:



Se demonstrază că:



Să se calculeze *fn*+1 într-o manieră Divide et Impera

8. Să se implementeze algoritmul lui Strassen (de la curs) pentru înmulțirea eficientă de matrice. (5p)