

**Tema 7**  
**28 noiembrie 2014**

***Probleme obligatorii***

**Termen de predare :** Laboratorul din săptămâna 11 (12 decembrie 2014)

**1. Arbori binari de căutare**

(1 p) **1.** Să se ordoneze descrescător un șir de cuvinte date de la tastatură, folosind un arbore binar de căutare.

(10 p) **2.** Sa se implementeze un *arbore binar de cautare echilibrat* cu urmatoarele operatii (cu echilibrare după fiecare operație, acolo unde este necesar) :

- (a) *adauga* (*t*, *x*) - insereaza cheia *x* in arborele de radacina *t*;
- (b) *cauta*(*t*, *x*) - intoarce 1 daca elementul *a* se afla in arborele de radacina *t* si 0 in caz contrar;
- (c) *maxim*(*t*) - intoarce elementul maxim din arborele de radacina *t*, fara a-l sterge din arbore;
- (d) *sterge*(*t*, *x*) - sterge in arborele de radacina *t* nodul cu cheia *x* (pastrand proprietatea de arbore binar de cautare și, eventual, echilibrarea);
- (e) *afiseaza* (*t*) – afișează cheile din arborele de rădăcină *t*, în ordine crescătoare

**2. Sortare**

(1 p) **3.** Să se implementeze metoda de ordonare Merge – Sort.

(2 p) **4.** Sa se implementeze algoritmul *randomized quick-sort* (alegerea pivotului se va face aleator).

***Probleme suplimentare***

**Termen de predare :** Laboratorul din săptămâna 11 (12 decembrie 2014)

(2 p) **5.** Sa se optimizeze procedura de *sortare rapidă*, folosind următoarea tehnică: subșirurile de dimensiune  $\leq 11$  elemente se sortează cu inserția directă.

(2 p) **6.** Să se optimizeze algoritmul de bază al metodei de *sortare prin interclasare* prin utilizarea inserției directe la sortarea subșirurilor mici (mai mici de 10 elemente).

(2 p) 7. Fie două secvențe sortate care împart același tablou și sunt poziționate astfel: prima crescând urmată de cealaltă descrescând, sau prima descrescând urmată de cealaltă crescând (secvență bitonică). Se cere să se sorteze prin *interclasare* tabloul respectiv.

(3 p) 8. Să se scrie algoritmul pentru sortarea unui șir de numere folosind metoda Heapsort. Structura de Heap va fi implementată ca un arbore binar într-una din cele două forme care urmează :

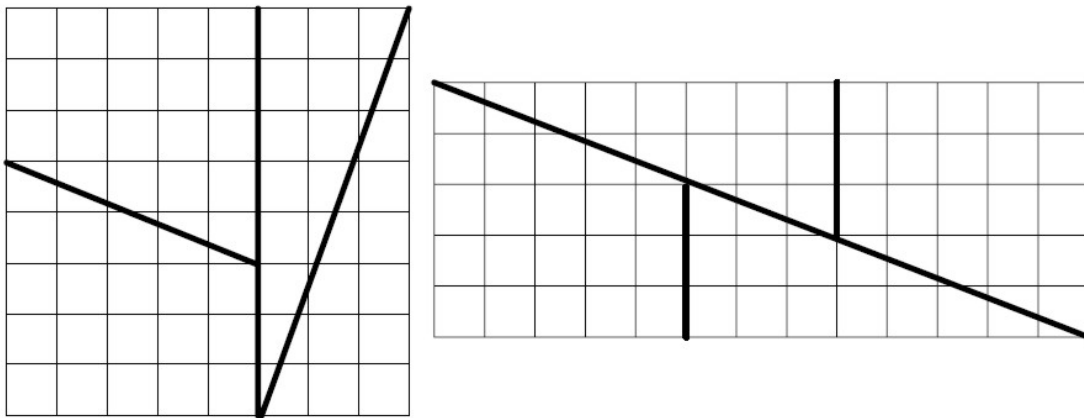
- a) max - Heap – arbore binar în care fiecare nod are cheia mai mare decât oricare dintre fiii săi
- b) min - Heap – arbore binar în care fiecare nod are cheia mai mică decât oricare dintre fiii săi

Scrieți funcții pentru crearea heap-ului și pentru decapitarea lui.

### Probleme facultative

**Termen de predare :** Laboratorul din săptămâna 11 (12 decembrie 2014)

(5 ps) 1. Fiind dată o tablă de șah de 8 x 8 pătrate, putem să o tăiem în două trapeze și două triunghiuri, ca în imaginea din stânga. O reasamblăm apoi după cum este indicat în figura din dreapta. Aria tablei din stânga este  $8 \times 8 = 64$ , pe când aria tablei din dreapta este  $13 \times 5 = 65$ . Explicați paradoxul.



Notând cu  $F(n)$  al  $n$ -lea număr Fibonacci, cum putem generaliza paradoxul? Găsiți o relație între  $F(n - 1)$ ,  $F(n)$  și  $F(n + 1)$  pentru a explica paradoxul.