

Proiectul I

1 Arbori binari de căutare AVL

Implementați o clasă `ArboreAVL` care sa ofere următoarele funcționalități:

- *Inserarea, ștergerea și căutarea unui element. 1p
- Parcurgere SRD(inordine), SDR(preordine), RSD(postordine). 1p
- Echilibrări ale arborilor (unde este cazul). 1p
- Supraîncărcarea operatorului `+` care să efectueze reuniunea a doi arbori (fără duplicate). 0.5p
- Supraîncărcarea operatorilor `<` și `>` care să efectueze compararea între elementele maxime a doi arbori. 0.4p
- Obținerea numărului de elemente din arbore. 0.1p

2 Set (mulțime)

Implementați o clasă `Set` (mulțime) care să ofere următoarele funcționalități:

- Fiecare element sa apară o singură dată.
- Dimensiune arbitrară a containerului. 1p
- *Inserarea, ștergerea și căutarea unui element. 1p
- Păstrarea elementelor în ordine crescătoare. 0.5p
- Supraîncărcarea operatorului `[]` pentru accesarea elementului de pe poziția `i`. 0.2p
- Supraîncărcarea operatorului `+` care să efectueze reuniunea a două mulțimi (fără duplicate). 0.5p
- Supraîncărcarea operatorilor `<` și `>` care să compare cardinalul a două mulțimi. 0.4p
- Obținerea elementelor pare, respectiv impare din mulțime. 0.3p
- Obținerea numărului de elemente din mulțime. 0.1p

3 Graf

Implementați o clasă Graf care să ofere următoarele funcționalități:

- Să se memoreze cât mai eficient structura de graf. 1p
- *Parcurgerea în lățime și lungime pornind dintr-un nod dat. 1p
- Distanța dintre două noduri date. 0.5p
- Supraîncărcarea operatorului [] care să întoarcă lista de adiacență a unui nod sub formă de vector. 0.5p
- Supraîncărcarea operatorilor < și > care să compare numărul de noduri și numărul de muchii a două grafuri (Fie G_1 și G_2 care conțin n_1 noduri și m_1 muchii, respectiv n_2 noduri și m_2 muchii. $G_1 > G_2$ dacă $n_1 > n_2$ sau $n_1 == n_2$ & $m_1 > m_2$). 0.4p
- Stabilirea dacă graful este arbore. 0.4p
- Obținerea numărului de noduri, respectiv muchii, ale grafului. 0.2p

4 Matrice

Implementați o clasă Matrice care să ofere următoarele funcționalități:

- *Supraîncărcarea operatorului + pentru adunarea a doua matrici. 0.4p
- Supraîncărcarea operatorului * pentru înmulțirea a doua matrici. 0.8p
- *Supraîncărcarea operatorului - pentru scăderea a doua matrici. 0.4p
- *Supraîncărcarea operatorului [] pentru a obținerea liniei i. 0.2p
- Calcularea determinantului matricii. 2p
- Obținerea numărului de linii, respectiv coloane. 0.2p

5 Polinoame

Implementați o clasă Polinom care să ofere următoarele funcționalități:

- *Calcularea valorii polinomului într-un anumit punct. 0.4p
- *Supraîncărcarea operatorului + pentru adunarea a două polinoame. 0.4p
- Supraîncărcarea operatorului * pentru înmulțirea a două polinoame. 0.8p
- Supraîncărcarea operatorului / pentru împărțirea a două polinoame. 2p
- *Supraîncărcarea operatorului [] pentru obținerea termenului cu gradul i. 0.2p
- Obținerea gradului polinomului. 0.2p

6 Cerințe globale

Următoarele cerințe sunt valabile pentru toate proiectele:

- Alocarea dinamică a memoriei. 0.5p
- Comentarii în cod care să ofere detalii despre modul de implementare. 0.7p
- Indentarea adecvată a codului. 0.4p
- Utilizarea unei convenții de denumire a variabilelor și a metodelor. 0.25p
- *Supraîncărcarea operatorilor de citire și afișare (<< și >>). 1p
- *Utilizarea constructorilor cu parametri, fără parametri, de copiere și supraîncărcarea operatorului =. 1.3p
- *Utilizarea destructorului. 0.5p
- *Separarea declarației de implementare a clasei/claselor. Declarația se va face în .h (header) și implementare în .cpp. 0.35p

7 Alte observații

- În cazul parcurgerilor rezultatul întors va fi un vector care conține elementele în ordinea de parcurgere (e.g. primul element parcurs pe prima poziție în vector, al doilea element pe a doua poziție în vector ș.a.m.d).
- Se acordă un punct din oficiu.
- Cerințele marcate cu * sunt necesare pentru a lua în considerare proiectul.
- Proiectul trebuie să compileze și să ruleze un mic demo care să demonstreze funcționalitățile cerute.
- Termen predare: vineri 21.03.2017 ora 23:59.
- Întârzierea cu o zi scade cu 2 puncte nota maximă obținută. Dacă se trimite proiectul între orele 00:00 - 23:59 ale primei zi după deadline se vor scădea 2p. Dacă se trimite proiectul între orele 00:00 - 23:59 ale zilei 2 după deadline se vor scădea 4p. Orice trimitere după mai mult de două zile va face ca proiectul să nu mai fie luat în considerare.
- Catalog: <https://1drv.ms/x/s!AmxvIYuN5gbogoAQPDtKKu7TxykOHg>