

Laborator 5

Exerciții

1. Preordine¹

Considerăm un arbore binar cu N noduri în care fiecare nod este numerotat de la 1 la N și conține o valoare număr natural. Să se afișeze valorile din arbore în urma parcurgerii în preordine (*rădăcină, stâng, drept*).

Date de intrare

Fișierul de intrare `preordine.in` conține pe prima linie numărul N . Fiecare dintre următoarele N linii conține câte 3 numere X `st` `dr`; linia $k + 1$ din fișier conține informațiile despre nodul numerotat cu k : X reprezintă valoarea din nod, `st` reprezintă numărul de ordine al descendentului stâng sau 0 dacă nodul k nu are descendent stâng, iar `dr` reprezintă numărul de ordine al descendentului drept sau 0 dacă nodul k nu are descendent drept.

Date de ieșire

Fișierul de ieșire `preordine.out` va conține pe prima linie N numere, separate prin exact un spațiu, reprezentând valorile din nodurile arborelui, obținute în urma parcurgerii în *preordine*.

Restricții

$$1 \leq N \leq 1.000$$

Valorile din nodurile arborelui vor fi mai mici sau egale cu 1.000.000.

2. Inordine²

Considerăm un arbore binar cu N noduri în care fiecare nod este numerotat de la 1 la N și conține o valoare număr natural. Să se afișeze valorile din arbore în urma parcurgerii în inordine (*stâng, rădăcină, drept*).

Date de intrare

Fișierul de intrare `inordine.in` conține pe prima linie numărul N . Fiecare dintre următoarele N linii conține câte 3 numere X `st` `dr`; linia $k + 1$ din fișier conține informații despre nodul numerotat cu k : X reprezintă valoarea din nod, `st` reprezintă numărul de ordine al descendentului stâng sau 0 dacă nodul k nu are descendent stâng, iar `dr` reprezintă numărul de ordine al descendentului drept sau 0 dacă nodul k nu are descendent drept.

¹ <https://www.pbinfo.ro/probleme/670/preordine>

² <https://www.pbinfo.ro/probleme/671/inordine>

Date de ieșire

Fișierul de ieșire `inordine.out` va conține pe prima linie N numere, separate prin exact un spațiu, reprezentând valorile din nodurile arborelui, obținute în urma parcurgerii în *inordine*.

Restricții

$$1 \leq N \leq 1.000$$

Valorile din nodurile arborelui vor fi mai mici sau egale cu 1.000.000.

3. Postordine³

Considerăm un arbore binar cu N noduri în care fiecare nod este numerotat de la 1 la N și conține o valoare număr natural. Să se afișeze valorile din arbore în urma parcurgerii în postordine (*stâng, drept, rădăcină*).

Date de intrare

Fișierul de intrare `postordine.in` conține pe prima linie numărul N . Fiecare dintre următoarele N linii conține câte 3 numere X st dr ; linia $k + 1$ din fișier conține informații referitoare la nodul numerotat cu k : X reprezintă valoarea din nod, st reprezintă numărul de ordine al descendentului stâng sau 0 dacă nodul k nu are descendent stâng, iar dr reprezintă numărul de ordine al descendentului drept sau 0 dacă nodul k nu are descendent drept.

Date de ieșire

Fișierul de ieșire `postordine.out` va conține pe prima linie N numere, separate prin exact un spațiu, reprezentând valorile din nodurile arborelui, obținute în urma parcurgerii în *postordine*.

Restricții

$$1 \leq N \leq 1.000$$

Valorile din nodurile arborelui vor fi mai mici sau egale cu 1.000.000.

4. CountSub⁴

Considerăm un arbore binar cu N noduri în care fiecare nod este numerotat de la 1 la N și conține o valoare număr natural. Se dau k noduri din arbore și se cere determinarea, pentru fiecare nod, a numărului de noduri din subarborele cu rădăcina în acel nod.

Date de intrare

Fișierul de intrare `countsub.in` conține pe prima linie numărul N . Fiecare dintre următoarele N linii conține câte 3 numere X st dr ; linia $i + 1$ din fișier conține informații referitoare la nodul numerotat cu i : X reprezintă valoarea din nod, st reprezintă numărul de ordine al descendentului stâng sau 0 dacă nodul i nu are descendent stâng, iar dr reprezintă numărul de ordine al descendentului drept sau 0 dacă nodul i nu are descendent drept.

³ <https://www.pbinfo.ro/probleme/672/postordine>

⁴ <https://www.pbinfo.ro/probleme/674/countsub>

Pe următoarea linie se află numărul k , iar pe fiecare dintre următoarele k linii se află câte un număr natural cuprins între 1 și N , reprezentând nodul curent.

Date de ieșire

Fișierul de ieșire `countsub.out` va conține k linii; fiecare linie va conține numărul de noduri din subarboarele cu rădăcina în nodul corespunzător.

Restricții

$$1 \leq N \leq 1.000, 1 \leq k \leq 1.000$$

Valorile din nodurile arborelui vor fi mai mici sau egale cu 1.000.000.

5. BiFrunze⁵

Considerăm un arbore binar cu N noduri în care fiecare nod este numerotat de la 1 la N și conține o valoare număr natural. Să se afișeze frunzele acestui arbore.

Date de intrare

Fișierul de intrare `bifrunze.in` conține pe prima linie numărul N . Fiecare dintre următoarele N linii conține câte 3 numere $X \ st \ dr$; linia $k + 1$ din fișier conține informații referitoare la nodul numerotat cu k : X reprezintă valoarea din nod, st reprezintă numărul de ordine al descendentului stâng sau 0 dacă nodul k nu are descendent stâng, iar dr reprezintă numărul de ordine al descendentului drept sau 0 dacă nodul k nu are descendent drept.

Date de ieșire

Fișierul de ieșire `bifrunze.out` va conține pe prima linie numerele de ordine ale nodurilor din arbore care sunt frunze, în ordine crescătoare, separate prin exact un spațiu.

Restricții

$$1 \leq N \leq 1.000$$

Valorile din nodurile arborelui vor fi mai mici sau egale cu 1.000.000.

6. CountPrimSub⁶

Considerăm un arbore binar cu N noduri în care fiecare nod este numerotat de la 1 la N și conține o valoare număr natural. Se dau k noduri din arbore și se cere determinarea, pentru fiecare nod, a numărului de noduri din subarboarele cu rădăcina în acel nod care conțin valori prime.

Date de intrare

Fișierul de intrare `countprimsub.in` conține pe prima linie numărul N . Fiecare dintre următoarele N linii conține câte 3 numere $X \ st \ dr$; linia $i + 1$ din fișier conține informațiile despre nodul numerotat cu i : X reprezintă valoarea nodului, st reprezintă numărul de ordine al

⁵ <https://www.pbinfo.ro/probleme/675/bifrunze>

⁶ <https://www.pbinfo.ro/probleme/676/countprimsub>

Algoritmica grafurilor

descendentului stâng sau 0 dacă nodul i nu are descendent stâng, iar dr reprezintă numărul de ordine al descendentului drept sau 0 dacă nodul i nu are descendent drept.

Pe următoarea linie se află numărul k , iar pe fiecare dintre următoarele k linii se află câte un număr natural cuprins între 1 și n , reprezentând nodul curent.

Date de ieșire

Fișierul de ieșire `countprimsub.out` va conține k linii; fiecare linie va conține numărul de noduri din subarboarele cu rădăcina în nodul corespunzător.

Restricții

$$1 \leq N \leq 1.000, 1 \leq k \leq 1.000$$

Valorile din nodurile arborelui vor fi mai mici sau egale cu 1.000.000.

7. NiveleBin⁷

Considerăm un arbore binar cu N noduri în care fiecare nod este numerotat de la 1 la N și conține o valoare număr natural. În acest arbore rădăcina este considerată pe nivelul 0, descendenții direcți ai rădăcinii pe nivelul 1 etc. Să se determine numărul de nivele k din arbore și, pentru fiecare nivel i de la 0 la k , numărul de noduri situate pe acel nivel.

Date de intrare

Fișierul de intrare `nivelebin.in` conține pe prima linie numărul N . Fiecare dintre următoarele N linii conține câte 3 numere X st dr ; linia $k + 1$ din fișier conține informații referitoare la nodul numerotat cu k : X reprezintă valoarea din nod, st reprezintă numărul de ordine al descendentului stâng sau 0 dacă nodul k nu are descendent stâng, iar dr reprezintă numărul de ordine al descendentului drept sau 0 dacă nodul k nu are descendent drept.

Date de ieșire

Fișierul de ieșire `nivelebin.out` va conține pe prima linie numărul k , iar pe a doua linie $k+1$ numere naturale separate prin exact un spațiu, al i -lea număr reprezentând numărul de noduri situate pe nivelul $i-1$ din arbore.

Restricții

$$1 \leq N \leq 1.000$$

Valorile din nodurile arborelui vor fi mai mici sau egale cu 1.000.000.

8. Înălțime1⁸

Se consideră un arbore binar în care nodurile memorează numere naturale nenule. Să se determine înălțimea arborelui. Înălțimea unui arbore este egală cu numărul de noduri de pe cel mai lung lanț elementar care unește rădăcina cu un nod terminal.

⁷ <https://www.pbinfo.ro/probleme/677/nivelebin>

⁸ <https://www.pbinfo.ro/probleme/761/inaltime1>

Date de intrare

Fișierul de intrare `inaltime1.in` conține pe prima linie lista valorilor memorate în nodurile arborelui, obținute în urma parcurgerii în preordine (*rădăcină, stâng, drept*). Dacă un nod nu are descendent stâng, în listă va apare valoarea 0. Dacă un nod nu are descendent drept, în listă va apare valoarea 0.

Date de ieșire

Fișierul de ieșire `inaltime1.out` va conține pe prima linie o valoare H , reprezentând înălțimea arborelui.

Restricții

Se recomandă folosirea arborilor alocați dinamic.

9. kNivel⁹

Se consideră un arbore binar în care nodurile memorează numere naturale nenule și un număr k . În arbore rădăcina se află pe nivelul 0, fii rădăcinii pe nivelul 1, fii fiilor rădăcinii pe nivelul 2 etc. Să se determine suma valorilor din nodurile aflate pe nivelul k .

Date de intrare

Fișierul de intrare `knivel1.in` conține pe prima linie lista valorilor memorate în nodurile arborelui, obținute în urma parcurgerii în preordine (*rădăcină, stâng, drept*). Dacă un nod nu are descendent stâng, în listă va apare valoarea 0. Dacă un nod nu are descendent drept, în listă va apare valoarea 0.

Linia a doua a fișierului conține numărul k .

Date de ieșire

Fișierul de ieșire `knivel1.out` va conține pe prima linie un singur număr S , reprezentând suma cerută.

Restricții

Se recomandă folosirea arborilor alocați dinamic.

10. BiLtime¹⁰

Se consideră un arbore binar în care nodurile memorează numere naturale nenule. Să se afișeze valorile din arbore în urma parcurgerii în lățime, pornind din rădăcină.

Date de intrare

Fișierul de intrare `bilatime.in` conține pe prima linie lista valorilor memorate în nodurile arborelui, obținute în urma parcurgerii în preordine (*rădăcină, stâng, drept*). Dacă un nod nu are descendent stâng, în listă va apare valoarea 0. Dacă un nod nu are descendent drept, în listă va apare valoarea 0.

⁹ <https://www.pbinfo.ro/probleme/760/knivel1>

¹⁰ <https://www.pbinfo.ro/probleme/759/bilatime>

Date de ieșire

Fișierul de ieșire `bilatime.out` va conține pe prima linie valorile din arbore în urma parcurgerii în lățime, pornind din rădăcină, separate prin câte un spațiu.

Restricții

Se recomandă folosirea arborilor alocați dinamic.

La parcurgerea în lățime, după vizitarea unui nod, se va vizita fiul stâng, apoi fiul drept.

11. Colegi¹¹

Un profesor vrea să știe care este cel mai frecvent prenume printre elevii din clasa noastră. Pentru aceasta a realizat o listă cu cele N prenume ale elevilor din clasă și acum vă cere să determinați prenumele cel mai frecvent și numărul său de apariții.

Dacă sunt mai multe prenume cu număr maxim de apariții se va determina primul în ordine alfabetică.

Date de intrare

Programul citește de la tastatură numărul N , iar apoi cele N prenume ale colegilor.

Date de ieșire

Programul va afișa pe ecran S , prenumele determinat.

Restricții

$$1 \leq N \leq 100.000$$

12. Binar¹²

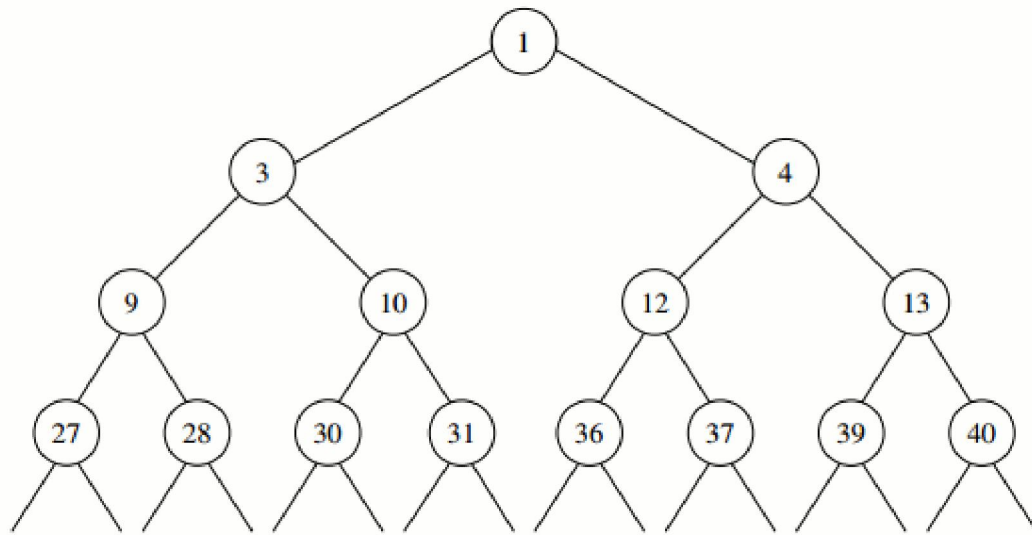
Fie p un număr întreg, mai mare decât 2. În nodurile unui arbore binar sunt scrise numere în modul următor:

- în rădăcina arborelui este scris 1.
- dacă în orice nod este scris x , copilul din stânga conține $p \times x$ și copilul din dreapta conține $p \times x + 1$.

De exemplu, dacă $p = 3$, arborele arată în felul următor:

¹¹ <https://www.pbinfo.ro/probleme/2268/colegi>

¹² <http://varena.ro/problema/binar>



Un număr se numește *drăguț* dacă acesta poate fi reprezentat în mod unic ca o sumă de două numere diferite care apar în nodurile arborelui.

Să se realizeze un program, care determină dacă numerele date n_1, n_2, n_3 și n_4 sunt *drăguțe*.

Date de intrare

Fișierul de intrare `binar.in` conține pe prima linie numerele întregi p, n_1, n_2, n_3 și n_4 .

Date de ieșire

În fișierul de ieșire `binar.out` se afișează pe prima linie, pentru fiecare din n_1, n_2, n_3 și n_4 , valoarea 1 dacă numărul este *drăguț* sau 0 dacă *nu este drăguț*.

Restricții

$2 < p < 50, 0 < n_1, n_2, n_3, n_4 < 1018$.