

Laborator 6

Exerciții

1. Heapsort

Se dă un șir cu N ($1 \leq N \leq 100.000$) numere întregi. Să se ordoneze descrescător valorile șirului cu ajutorul unui *min-heap*.

Date de intrare

Fișierul de intrare `heapsort.in` conține pe prima linie numărul N , iar pe următoarea linie N numere naturale, separate prin exact un spațiu, reprezentând valorile elementelor șirului.

Date de ieșire

Fișierul de ieșire `heapsort.out` va conține pe o linie N numere naturale, separate prin exact un spațiu, reprezentând valorile elementelor șirului ordonate descrescător.

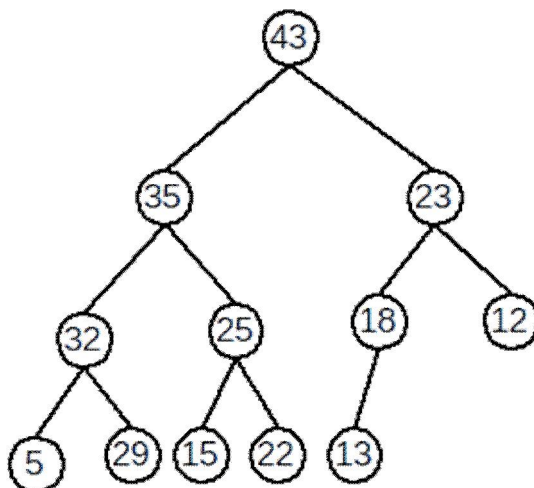
2. Heap¹

Un *max-heap* este un arbore binar complet cu următoarea proprietate suplimentară: valoarea din orice nod este mai mare sau egală cu valorile din orice nod descendent.

Similar se definește noțiunea de *min-heap*: valoarea din orice nod este mai mică sau egală cu valorile descendenților.

Într-un *max-heap* rădăcina are valoare maximă, iar într-un *min-heap* rădăcina are valoare minimă. Nu se precizează nicio relație între valorile din fiii unui nod.

Arborele binar complet din figura următoare este un *max-heap*:



¹ <https://www.pbinfo.ro/probleme/1855/heap>

Algoritmica grafurilor

Pentru ca un arbore binar complet să fie *max-heap* (similar pentru *min-heap*), fiecare nod din arbore trebuie:

- să fie mai mare sau egal cu descendenții săi (dacă există);
- să fie mai mic sau egal cu tatăl său (dacă există).

Să presupunem că un arbore binar complet $H[]$ are proprietatea de *max-heap*, cu excepția unui nod k . Cum îl corectăm, astfel încât să devină *max-heap*?! Distingem două cazuri:

- dacă nodul k este mai mare decât tatăl său ($H[k] > H[k/2]$) îl vom muta în sus în arbore, până când acesta devine *max-heap*. Această operație se numește *promovare* a unui nod în heap;
- dacă nodul k este mai mic decât cel puțin unul dintre fii săi ($H[k] < H[2*k]$ și / sau $H[k] < H[2*k+1]$) îl vom muta în jos în mod convenabil, până când arborele devine *max-heap*. Această operație se numește *cernere* a unui nod în arbore.

Se consideră o colecție de numere naturale, inițial vidă. Asupra ei se fac două tipuri de operații:

- 1 x – valoarea x se adaugă în colecție;
- 2 – cea mai mare valoare din colecție se afișează, apoi se elimină din colecție.

Dându-se un șir de m operații, să se afișeze în ordine rezultatele operațiilor de tip 2.

Date de intrare

Fișierul de intrare `heap.in` conține pe prima linie numărul m , iar pe următoarele m linii câte o operație.

Date de ieșire

Fișierul de ieșire `heap.out` va conține rezultatele operațiilor de tip 2, câte unul pe o linie, în ordinea în care au fost efectuate.

Restricții

$$1 \leq m \leq 250.000, 1 \leq x \leq 1.000.000.000.$$

3. Competiție²

K concurenți participă la o întrecere. Fiecare dintre ei trebuie să facă N tururi. Toți concurenții pleacă în același timp de la linia de pornire. La început fiecare concurent se simte normal, însă când termină o tură el își pierde din capacitatea de a conduce. Astfel fiecare cursă este cu o milisecundă mai înceată decât cea dinainte. La început concurentul i face un tur în ms_i milisecunde (ms_i este un număr natural nenul). Regulile spun ca fiecare concurent i are dreptul la fiecare p_i ($1 \leq p_i \leq N$) să primească, când trece linia de sosire, o băutură energizantă ce îl readuce la forma normală. Fiecare concurent va trece linia de sosire de N ori (vom număra ultima trecere, însă nu și pe cea de la început).

² <http://varena.ro/problema/competitie>

Date de intrare

Pe prima linie din fișierul `competitie.in` se vor găsi două numere: K și N . Pe următoarele K linii se găsesc două numere ms_i și p_i .

Date de ieșire

Fișierul `competitie.out` va conține numărul maxim de concurenți ce trec linia de start în același timp.

Restricții

$$K \leq 10.000, N \leq 1.000, ms_i \leq 1.000.000, 1 \leq p_i \leq N.$$

4. Frunze³

Se dă vectorul de tați al unui arbore cu rădăcină cu N noduri. Determinați rădăcina arborelui și frunzele acestuia.

Date de intrare

Fișierul de intrare `frunze.in` conține pe prima linie numărul de noduri N . Pe linia următoare se află vectorul de tați al arborelui, valorile fiind separate prin spații.

Date de ieșire

Fișierul de ieșire `frunze.out` va conține pe prima linie rădăcina arborelui. A doua linie va conține numărul de frunze din arbore, iar următoarea linie frunzele, în ordine crescătoare și separate printr-un spațiu.

Restricții

$$1 \leq N \leq 100.$$

În vectorul de tați, rădăcina este marcată cu 0.

5. Nivele⁴

Într-un arbore cu rădăcină, spunem că rădăcina este pe nivelul 1, fiii rădăcinii pe nivelul 2, fiii fiilor rădăcinii pe nivelul 3 etc.

Se dă vectorul de tați al unui arbore cu rădăcină cu N noduri și k noduri din arbore. Determinați pentru fiecare dintre cele k noduri nivelul pe care se află.

Date de intrare

Fișierul de intrare `nivele.in` conține pe prima linie numărul de noduri N . Pe linia următoare se află vectorul de tați al arborelui, valorile fiind separate prin spații. Linia a treia conține numărul k , iar linia a patra k noduri, $x[1], x[2], \dots, x[k]$.

³ <https://www.pbinfo.ro/probleme/637/frunze>

⁴ <https://www.pbinfo.ro/probleme/638/nivele>

Date de ieșire

Fișierul de ieșire `nivele.out` va conține k linii. Linia i va conține nivelul pe care se află nodul $x[i]$ în arborele dat

Restricții

$$1 \leq k \leq N \leq 100$$

În vectorul de tați rădăcina este marcată cu 0.

6. Înălțime⁵

Într-un arbore cu rădăcină, spunem că rădăcina este pe nivelul 1, fiii rădăcinii pe nivelul 2, fiii fiilor rădăcinii pe nivelul 3 etc. Numărul de nivele distincte din arbore determină înălțimea arborelui.

Se dă vectorul de tați al unui arbore cu rădăcină cu N noduri. Determinați înălțimea arborelui.

Date de intrare

Fișierul de intrare `inaltime.in` conține pe prima linie numărul de noduri N . Pe linia următoare se află vectorul de tați al arborelui, valorile fiind separate prin spații.

Date de ieșire

Fișierul de ieșire `inaltime.out` va conține pe prima linie numărul H , reprezentând înălțimea arborelui.

Restricții

$$1 \leq N \leq 100$$

În vectorul de tați, rădăcina este marcată cu 0.

7. Subarbore⁶

Se dă vectorul de tați al unui arbore cu rădăcină cu N noduri și un nod k . Afișați, în ordine crescătoare, nodurile din subarboarele cu rădăcina în k .

Date de intrare

Fișierul de intrare `subarbore.in` conține pe prima linie numărul de noduri N și nodul k . Pe linia următoare se află vectorul de tați al arborelui, valorile fiind separate prin spații.

Date de ieșire

Fișierul de ieșire `subarbore.out` va conține pe prima linie nodurile din subarbore, în ordine crescătoare, separate printr-un spațiu.

⁵ <https://www.pbinfo.ro/probleme/639/inaltime>

⁶ <https://www.pbinfo.ro/probleme/641/subarbore>

Restricții

$$1 \leq k \leq N \leq 100$$

În vectorul de tați rădăcina este marcată cu 0.

8. DetDrum⁷

Se dă vectorul de tați al unui arbore cu rădăcină cu N noduri și un nod k . Determinați drumul de la nodul k la rădăcina arborelui.

Date de intrare

Fișierul de intrare `detdrum.in` conține pe prima linie numărul de noduri N și nodul k . Pe linia următoare se află vectorul de tați al arborelui, valorile fiind separate prin spații.

Date de ieșire

Fișierul de ieșire `detdrum.out` va conține pe prima linie nodurile care alcătuiesc drumul determinat, separate printr-un spațiu.

Restricții

$$1 \leq k \leq N \leq 100$$

În vectorul de tați, rădăcina este marcată cu 0.

9. DetDrum2⁸

Se dă vectorul de tați al unui arbore cu rădăcină cu N noduri și două noduri p și q . Determinați drumul elementar de la nodul p la nodul q .

Date de intrare

Fișierul de intrare `detdrum2.in` conține pe prima linie numărul de noduri N și nodurile p și q . Pe linia următoare se află vectorul de tați al arborelui, valorile fiind separate prin spații.

Date de ieșire

Fișierul de ieșire `detdrum2.out` va conține pe prima linie nodurile care alcătuiesc drumul determinat, separate printr-un spațiu.

Restricții

$$1 \leq k \leq N \leq 100$$

În vectorul de tați, rădăcina este marcată cu 0.

⁷ <https://www.pbinfo.ro/probleme/642/detdrum>

⁸ <https://www.pbinfo.ro/probleme/644/detdrum2>

10. Arbore⁹

Se dă un arbore reprezentat prin vectorul de tați. Arborele are N noduri, numerotate de la 1 la N . Se cere să se afișeze frunzele arborelui și descendenții a t noduri.

Date de intrare

Fișierul de intrare **arbore.in** numerele naturale N și t . Pe următoarea linie se află vectorul de tați al arborelui. Pe cea de-a treia linie se află t numere separate prin câte un spațiu, ce reprezintă indicii nodurilor ai căror descendenți trebuie afișați.

Date de ieșire

În fișierul de ieșire **arbore.out** se vor afișa pe prima linie frunzele arborelui, în ordine crescătoare. Pe următoarele t linii se vor afișa, separați prin câte un spațiu, descendenții nodurilor cerute, în ordine crescătoare.

Restricții

$$1 \leq N \leq 2.000, 1 \leq t \leq 20.$$

11. Bomboane¹⁰

Bianca și Maria au împreună o cutie de bomboane. Întrucât Biancăi nu îi prea plac bomboanele, aceasta a inventat următorul joc. Ea îi oferă Mariei un clasament al bomboanelor. Acest clasament este de forma unui arbore cu rădăcină. Bomboanele sunt numerotate de la 1 la N , iar Bianca îi spune Mariei pentru fiecare bomboană, care este bomboana mai bună decât ea. După aceea, Bianca îi pune t întrebări Mariei de tipul: *Pe ce nivel se află bomboana x ?*

Pentru fiecare răspuns corect, Maria primește cadou de la Bianca acea bomboană. Mariei îi este frică să nu greșească și astfel, să piardă o bomboană, și vă roagă pe voi să o ajutați să răspundă corect la întrebările Biancăi.

Date de intrare

Fișierul de intrare **bomboane2.in** conține pe prima linie numerele naturale N și t . Pe cea de-a doua linie se află clasamentul bomboanelor reprezentat în felul următor: pentru bomboana cu indicele k se știe indicele bomboanei mai bune decât ea. Pe cea de-a treia linie se află t numere ce reprezintă indicii bomboanelor ale căror nivel în clasament trebuie să-l afle Maria.

Date de ieșire

Fișierul de ieșire **bomboane2.out** va conține t linii. Pe fiecare linie k se va afișa nivelul pe care se află a k -a bomboană din cele t din întrebările Biancăi.

Restricții

$$1 \leq N \leq 2.000, 1 \leq t \leq 20.$$

Bomboana care se află în vârful clasamentului nu va avea nicio altă bomboană mai bună decât ea și i se va atribui indicele 0 în fișierul de intrare. Ea se va numi rădăcina clasamentului.

⁹ <http://varena.ro/problema/arbore>

¹⁰ <http://varena.ro/problema/bomboane2>

Numerotarea nivelurilor se va începe de la nivelul următor rădăcinii.

12. Ierarhie¹¹

Andrei tocmai s-a angajat într-o firmă. Pentru a nu face nicio confuzie între angajați, el și-ar dori să cunoască ierarhia din acea firmă. Cu toate acestea, atunci când s-a angajat, lui Andrei i s-a dat doar o listă cu șeful direct al fiecărui angajat. Andrei nu se prea descurcă cu această listă. Știind că angajații sunt numerotați de la 1 la N, el vă roagă să îl ajutați să afle atât cine este șeful tuturor, cât și care sunt subordonații direcți ai fiecărui angajat.

Date de intrare

Fișierul de intrare `ierarhie.in` conține pe prima linie numărul natural N, ce reprezintă numărul de angajați. Pe cea de-a doua linie se află N numere cu semnificația că numărul de pe poziția k este șeful direct al angajatului k.

Date de ieșire

În fișierul de ieșire `ierarhie.out` se va afișa pe prima linie șeful tuturor angajaților. Pe următoarele linii se va afla numărul angajatului și subordonații săi direcți, afișați în ordine crescătoare separați prin câte un spațiu. Dacă un angajat nu are niciun subordonat direct, se va trece la angajatul următor.

Restricții

$$1 \leq N \leq 2.000$$

Dacă un angajat nu are niciun șef direct, atunci acestuia i se va atribui numărul 0.

13. Prepost¹²

Fie un arbore general cu rădăcină cu N noduri numerotate de la 1 la N. Se cunosc parcurgerile sale în preordine și în postordine, date sub forma a câte unui vector cu N elemente. Să se reconstituie muchiile arborelui, indicând pentru fiecare nod cine este părintele lui.

Date de intrare

Fișierul de intrare `prepost.in` conține pe prima linie numărul de noduri N. Pe a doua linie se găsește o permutare a mulțimii $\{1, 2, \dots, N\}$ indicând parcurgerea în preordine. Pe a treia linie se găsește o altă permutare a mulțimii $\{1, 2, \dots, N\}$ indicând parcurgerea în postordine.

Date de ieșire

În fișierul de ieșire `prepost.out` se vor tipări N numere. Al k-lea număr indică părintele nodului k. Pentru rădăcină se va tipări valoarea 0.

Restricții

$$1 \leq N \leq 100.000.$$

¹¹ <http://varena.ro/problema/ierarhie>

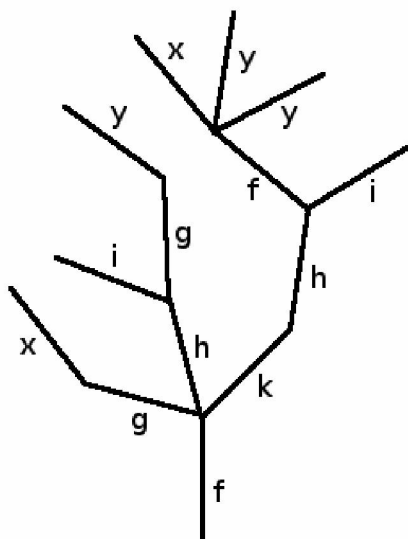
¹² <http://varena.ro/problema/prepost>

14. Pom¹³

În grădina lui Ion a crescut peste noapte Pomul de Halloween, fratele cel rău al Pomului de Crăciun. Ion, fiind șomer și având mult timp liber, a studiat copacul și a observat că există doar N tipuri de ramuri, pe care le-a codificat prin litere mici ale alfabetului (ca toți românii, Ion nu folosește diacritice). Toate ramurile de același tip t se despart în alte $R[t]$ ramuri. Ion a codificat arborele astfel:

- O ramură terminală se codifică doar prin tipul ei.
- O ramură interioară se codifică prin tipul ei, apoi, între paranteze și despărțite prin virgule, codificările subarborilor.
- Ramurile sunt întotdeauna enumerate de la stânga la dreapta.

Pentru imaginea alăturată, $R[f] = 3, R[g] = 1, R[h] = 2, R[i] = 0, R[k] = 1,$
 $R[x] = 0, R[y] = 0,$ iar codificarea arborelui este
 $f(g(x), h(i, g(y)), k(h(f(x, y, y), i)))$.



Apoi, Ion s-a apucat să taie copacul, ca să aibă lemne de foc. Ca să nu se prindă Primăria, Ion taie doar ramuri terminale, una câte una. Când are mai multe variante, o alege pe care apare prima în codificare. Să se afle ce tipuri de ramuri va obține Ion, în ordine.

Date de intrare

Fișierul de intrare `pom.in` conține, pe prima linie, numărul N de tipuri de ramuri. Următoarele N linii conțin tipul unei ramuri, t_i și numărul de ramuri $R[t_i]$ în care se ramifică acest tip, separate printr-un spațiu. Ultima linie a fișierului conține codificarea arborelui. Aceasta poate conține litere mici, paranteze rotunde, virgule și spații. Spațiile trebuie ignorate. Fișierul se termină cu un caracter `'\n'` (linie nouă).

¹³ <http://varena.ro/problema/pom>

Date de ieșire

În fișierul de ieșire `pom.out` se vor scrie, lipite, tipurile de ramuri în ordinea tăierii. Linia va fi încheiată cu un caracter `'\n'` (linie nouă). Dacă Ion și-a notat greșit codificarea arborelui, în fișier trebuie scris numărul `-1`.

Restricții

$1 \leq N \leq 26$; $'a' \leq t_i \leq 'z'$ pentru $1 \leq i \leq N$; $0 \leq R[t_i] \leq 100.000$ pentru $1 \leq i \leq N$.

Lungimea codificării, cu spații cu tot, este cel mult `100.000`.

Toate cele N tipuri de ramuri sunt distincte.