Problema Bomboane

Se foloseste un min heap pentru minime și un max heap pentru maxime

Citim datele din fisier si in acelasi timp calculam nr total de bomboane si introducem datele in heap-uri

Daca suma (nr total bomboane) nu e divizibila cu n atunci nu e posibil sa impartim uniform bomboanele

Calculam Nr mediu de bomboane, la final fiecare cutie va contine acest numar de bomboane

Efectuam operatii de mutare a bomboanelor doar daca au mai ramas cutii cu nr de bomboane mai mare decat media, altfel ne oprim

Calculam de cate bomboane mai are nevoie minimul curent

Calcuam si cate bomboane mai putem sa mai luam de la maxim, pentru ca nu putem / nu ar fi eficient sa scadem subnr\_mediu\_bomboane

Aici distingem 3 cazuri

1. Exista mai multe bomboane decat e suficient, caz in care reintroducem in heap doar maximul cu nr de bomboane actualizat

2. Exista bomboane fix atat cat ar fi nevoie (maximul si minimul curent sunt egal departate de medie). In acest caz nu mai reintroducem nimic in heap

3. In maxim exista un numar de bomboane mai mic decat este necesar, prin urmare trebuie să reintroducem in heap minimul

Problema cifre4

Folosim un vector de frecventa de lungime P-1 ce retine daca restul r a fost gasit pana in momentul de fata

Folosim o coada in care se introduc numere de "candideaza" pt rezultat, initial vom introduce în ea 0. Repetam procesul pana cand avem un element cu proprietatea gasita

Daca acelasi rest a mai fost obtinut anterior nu mai are sens sa continuam pentru ca avem deja un numar mai mic care a obtinut acelasi rest (asta verificam cu ajutorul vectorului de frecventa)

Construim noii candidati pe care ii adaugam in coada, adugand la candidatul curent ca si sufix fiecare dintre cifrele eligibile, in ordine crescatoare:

for (int i = 0; i < 4; i++)

{

q.push(x \* 10 + cifre[i]);

}

Problema lastk

Se foloseste un min heap pentru a stoca numerele din sirul generat

Se genereaza elementul curent din sir

Introducem in min heap primele k elemente

Afisam cele mai mari k elemente in ordine crescatoare, folosind pop

Problema dreptunghiuri5

Se da o matrice cu N linii si M coloane plina cu 0 si 1.

Se cere sa se afle cate dreptunghiuri maximale exista.

Un dreptunghi maximal este o submatrice a matricii date care este plina cu 0 si are proprietatea ca nu se mai poate extinde.

O submatrice se poate extinde daca cel putin una din laturile ei are ca vecini doar casute cu 0.

Citim elementele matricei

Pentru fiecare coloana, nrZeroDeasupra[i] o sa retina cate zerouri sunt deasupra liniei curente (inclusiv linia curenta) inainte de a aparea un 1

Deci prin intermediu acestui vector putem sa ne dam seama cat de mult ne putem extinde cum submatricea pe verticala

Daca nrZeroDeasupra[i] = 5, atunci inseamna ca putem forma o submatrice cu inaltimea maxim 5 avand ca ultima linie i si ca prima linie i-4

Calculam nrZerouriDeasupra pentru fiecare coloana in parte

Vectorul este initializat o singura data in afara loop-ului de parcurgere a liniilor deoarece la fiecare linie se aduna 1 sau se revine la 0 (in functie de valoarea gasita in matrice)

Daca apare un 1, resetam numarul de zerouri care se afla deasupra liniei curente

Altfel crestem numarul de zerouri

Folosim o stiva in care vom retine, pentru linia curenta, toate coloanele pe care le vom procesa pentru a vedea pana unde ne putem "extinde" cu submatricea curenta de zerouri pe orizontala

Informatia legata de cat de mult ne putem extinde pe verticala (avand ca linie de finish linia curenta) o avem deja pentru fiecare coloana in parte in vectorul nrZeroDeasupra

Parcurgem fiecare coloana in parte, si mergem cu j pana la nr\_coloane + 1 pentru a acoperi si cazul special in care submatricea maximala de zerouri se intinde pe orizontala pana la ultima coloana

Scoatem din stiva toate coloanele deja existente pentru care exista un numar de zerouri desupra mai mare decat numarul de zerouri de deasupra coloanei curente

Pentru fiecare element pe care il scoatem, verificam daca acesta contribuie la formarea unei submatrice maximale

Daca lastok e mai mic jOprire incrementam numarul de submatrice maximale

Scoatem din stiva toate coloanele deja existente pentru care exista un numar de zerouri desupra egal cu numarul de zerouri de deasupra coloanei curente

Introducem in stiva coloana curenta

Daca dedesubt avem un i, jOprire va fi setat la j nu mai putem avansa deoarece (nu putem construi o submatrice mai mare)