

## Cifru - solutie

Pentru a determina valorile care trebuie adunate in vederea obtinerii sumei  $S$  trebuie determinate toate submultimile multimii cu  $n$  elemente. Numarul acestor submultimumi este  $2^n$ . Ceea ce pentru  $n=50$  devine practic imposibil din punctul de vedere al factorului timp.

Tinand cont insa de faptul ca trebuie calculata o suma, aceasta se poate calcula ca suma a doua sume partiale:

$$S_n = S_{\lfloor n/2 \rfloor} + S_{n - \lfloor n/2 \rfloor}$$

Aceasta observatie are ca efect faptul ca trebuie determinate doar  $2^{\lfloor n/2 \rfloor}$  submultimumi, adica maxim  $2^{25}$  submultimumi.

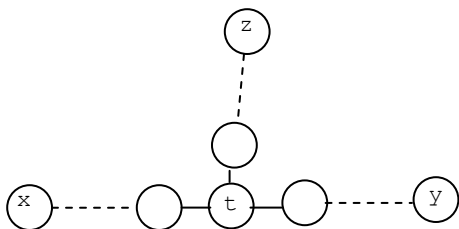
Aceste submultimumi se determina si sumele corespunzatoare se ordoneaza.

Pentru celelalte  $2^{n - \lfloor n/2 \rfloor}$  submultimumi se determina pe rand sumele respective ( $S_i$ ),  $0 \leq i \leq n - \lfloor n/2 \rfloor$  si se cauta (cautare binara) in sumele retinute anterior daca nu cumva exista o valoare care completeaza suma ( $S - S_i$ ). In cazul in care aceasta valoare exista, solutia a fost determinata.

## Copac - solutie

Problema se rezolva in doua etape. In prima etapa se determina o relatie intre frunze corespunzatoare unei parcurgeri Euler valide a arborelui. Altfel spus, in aceasta parcurgere nu vor exista interclasate frunze apartinand subarborilor disjuncti ai unui nod al arborelui. Pentru orice subarbore, multimea frunzelor care apartine acestuia va forma un interval compact. Aceasta ordine se determina folosind  $O(N \lg N)$  interogari.

Se porneste cu 3 frunze aranjate intr-o lista intr-o ordine oarecare, apoi se insereaza pe rand fiecare dintre frunzele ramase folosind o cautare binara.



Alegand trei frunze  $x, y, z$  se folosesc 3 interogari pentru a determina distanta de la nodul  $z$  pana la nodul  $t$ , unde lantul ce leaga nodul  $x$  de nodul  $z$ , respectiv cel ce leaga nodul  $y$  de nodul  $z$  "parasesc" lantul dintre  $x$  si  $y$  ( $d_{zt} = (d_{xz} + d_{yz} - d_{xy})/2$ ). Fixand trei frunze  $l, m$  si  $r$  (aranjate in lista in aceasta ordine) se determina daca un nou nod se afla intre  $l$  si  $m$  sau intre  $m$  si  $r$ . Se procedeaza astfel pentru toate nodurile, fiecare inserare realizandu-se folosind  $O(\lg N)$  interogari.

In a doua etapa se construiesc efectiv arborele. Se construiesc drumul dintre primele doua noduri aflate in lista, apoi pe rand se adauga urmatoarea frunza din lista generata la pasul anterior. La

fiecare pas se mentine o stiva continand nodurile de pe lantul de la primul nod din lista pana la ultima frunza inserata, care este actualizata dupa inserarea noii frunze, impreuna cu nodurile care o leaga de lant.

Pentru a ne asigura ca nu se vor pune intrebari redundante, toate raspunsurile primite se pastreaza intr-un hash.

Pentru detalii consultati sursa oficiala. Aceasta nu foloseste niciodata mai mult de 9100 de interogari pe datele de test.

## **Harta - solutie**

Pentru a simplifica implementarea, tinand cont de faptul ca toate dreptunghiurile au dimensiunea  $k \times 3k$ , toate ordonatele se inmultesc cu 3. Astfel dreptunghiurile se transforma in patrate iar in implementare nu mai este nevoie de numere reale.

Este intuitiv ca solutia se poate gasi folosind cautarea binara. Astfel, pentru o lungime  $k$  fixata a laturii se doreste sa se determine in  $O(N)$  daca patratele avand colturile de stanga-sus in punctele date si de latura  $k$  se intersecteaza.

Se construiesc un caroiaj constand din celulele de latura  $k$  determinate de toate punctele de coordonate  $(x*k, y*k)$  cu  $x$  si  $y$  intregi. Evident, daca doua puncte apartin aceleiasi celule, patratele determinate de cele doua puncte se intersecteaza si trebuie cautat un  $k$  mai mic. De asemenea, este suficient sa mai verificam doar intersectiile intre patratele determinate de puncte apartinand celulelor vecine ale caroiajului. Din moment ce fiecare celula va contine maxim un punct (cautarea oprindu-se in cazul in care se vor gasi doua apartinand aceleiasi celule) si fiecare celula are 8 vecini, verificarea se face in  $O(1)$  pentru fiecare dintre puncte. Pentru a verifica rapid daca celula alaturata unei celule fixate contine puncte, se recomanda folosirea unui hash.