

SUPRAINCARCAREA OPERATORILOR

Exercitiu 1

Partea 1

Sa se defineasca o clasa "bitset" pentru o multime de numere intregi reprezentata printr-un vector de biti. Bitul din pozitia "k" este 1 daca multimea contine numarul "k" si este 0 daca multimea nu contine pe "k".

Exemplu: multime de max 8 elemente cu continut [1,3,6,7] :

11001010

Multimea poate avea orice numar de elemente, vectorul este alocat dinamic dar nu se mai poate extinde peste dimensiunea data la construirea multimii.

Date:

- vector de intregi (dimensiune minima 16 elemente = un intreg)
- dimensiune vector (numar de elemente sau numar de biti)

Metode:

- Constructor cu parametru dimensiune multime (implicit 256) (creaza o multime vida, deci contine numai zerouri)
- Destructor (elibereaza memoria)
- "contains": Test de apartenenta la multime a unei valori date
- "contains": Test de includere a unei multimi in multimea "\*this"
- "isEmpty": Test daca multime vida
- "size" : cardinal multime (numar de elemente)

Operatori:

- Afisare elemente multime, intre paranteze drepte ( << )
- Adaugarea unui element la o multime (+) (functie membru)
- Reuniunea a doua multimi, fara modificare operanzi (+)
- Diferenta a doua multimi, fara modificare operanzi (-)
- Intersectia a doua multimi, fara modificare operanzi (\*)
- Atribuirea intre multimi (=)
- Comparatia a doua multimi la egalitate (==)

Indicatii:

- Bitul "k" se afla in pozitia "j" din  $m[j]$  , daca "m" este numele vectorului de intregi din clasa. Indicii i si j se calculeaza astfel:

$$i = k / 16$$

$$j = k \% 16$$

## Partea a 2-a

Utilizarea clasei "bitset".

Dupa verificarea operatiilor clasei "bitset" se va scrie un program pentru acoperirea unei multimi scop S cu un numar minim de multimi dintr-o colectie de multimi C (un vector de obiecte "bitset"), folosind un algoritm "greedy". Exemplu de date si rezultate:

Multimea scop  $S = [1, 2, 3, 4, 5]$

Colectia :  $C[1] = [2, 5]$ ,  $C[2] = [2, 3, 4]$ ,  $C[3] = [3, 4, 5]$ ,  $C[4] = [1, 3, 4]$

Solutia greedy:  $C[2], C[1], C[4]$  (solutia optima  $C[1], C[4]$  )

Algoritmul greedy:

verifica daca  $C[1] + C[2] + \dots + C[n]$  contine pe S (daca exista solutie)

repete cat timp S nu e vida

repete pentru i de la 1 la n

$A[i] = S * C[i]$  (intersectie)

retine in imax numar multime  $A[i]$  cu cardinal maxim

afiseaza numarul si continutul multimii  $C[imax]$

$S = S - C[imax]$

Multimile  $C[i]$  si S vor fi create prin adaugari succesive de elemente, generate prin program sau citite de la consola.