

sqtpm

[186154]

[voltar](#)**Trabalho:** 99-bits

Linguagens: C

Data de abertura: 2017/09/04 14:00:00

Data limite para envio: 2017/10/04 18:00:00

Número máximo de envios: 25

Casos-de-teste abertos: [casos-de-teste.tgz](#)

Conjunto em vetor de bits

Uma forma de representar um subconjunto S de $\{1, 2, \dots, n\}$ é usando um vetor de bits V . Representado dessa forma, se o elemento i está em S então o bit i de V é igual a 1, caso contrário é igual a 0.

Essa representação usa sempre n bits para um subconjunto S de $\{1, 2, \dots, n\}$. Uma outra possibilidade é usar um vetor e armazenar nele apenas os elementos que estão em S , o que vai exigir $|S| \cdot \lg n$ bits. Se S tem muitos elementos, então a representação usando vetor de bits será vantajosa.

Neste trabalho você deve representar um subconjunto de $\{1, 2, \dots, n\}$ como um vetor de bits.

Entrada

A entrada é composta por uma sucessão de comandos, um por linha. Os possíveis comandos estão descritos abaixo.

- `create n`

Cria um subconjunto vazio de $\{1, 2, \dots, n\}$. Vamos chamá-lo de S . Se já existe um subconjunto ele deve ser destruído e um novo subconjunto deve ser criado.

- `add i`

Adiciona o elemento i a S .

- `remove i`

Remove o elemento i de S .

- `in i`

Testa se o elemento i está em S . Imprime "belongs(i) = " seguido de "true" se ele estiver ou "false" se ele não estiver.

- `rank i`

sqtpm

[186154]

[voltar](#)

Imprime "rank(i) = " e o número de elementos em S que são menores ou iguais a i. Se i for maior que n, o programa deve imprimir "rank(i) = 0".

- **select i**

Imprime "select(i) = " e o i-ésimo menor elemento em S. Se i for maior que |S|, o programa deve imprimir "select(i) = 0".

- **rangecount j k**

Imprime "rangecount(j,k) = " e o número de elementos em S no intervalo [j,k].

- **print**

Imprime "S = " e os elementos em S em ordem crescente em uma única linha, dentro de chaves e separados por vírgulas.

- **exit**

Termina o programa.

Sugere-se que as operações sejam implementadas diretamente para treinar a manipulação de bits e não apenas fazendo uma chamada às macros set e test que foram vistas em aula.

Saída

A saída deve conter as linhas geradas pelos comandos na entrada.

Exemplo:

Entrada:

```
create 10
add 1
add 3
add 5
add 7
add 9
remove 9
print
in 9
rank 5
rank 6
rank 7
```

sqtpm

[186154]

[voltar](#)

```
rangecount 2 6  
exit
```

Saída:

```
S = {1,3,5,7}  
belongs(9) = false  
rank(5) = 3  
rank(6) = 3  
rank(7) = 4  
select(3) = 5  
select(4) = 7  
rangecount(2,6) = 2
```
