SCC 202 - Algoritmos e Estruturas de Dados I

TAD: Tipo Abstrato de Dados (3)
Prática

10/8/2010

Exercício: TAD Conjuntos (SET)

- Um conjunto é uma coleção de membros (ou elementos); cada membro ou é um conjunto ou um elemento primitivo chamado de átomo.

- Todos os membros são diferentes: nenhum conjunto contém 2 cópias do mesmo elemento.

- Ex:

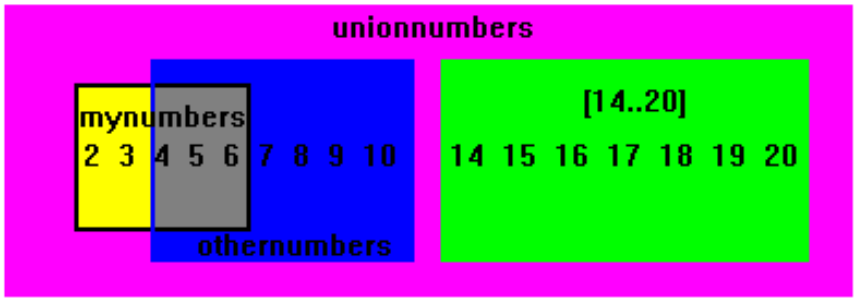
{1,4} ok

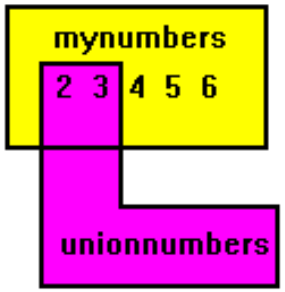
{1,4,1} não ok

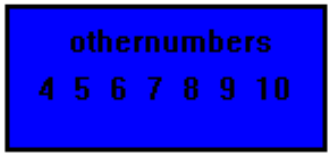
Operações básicas: união, intersecção e diferença

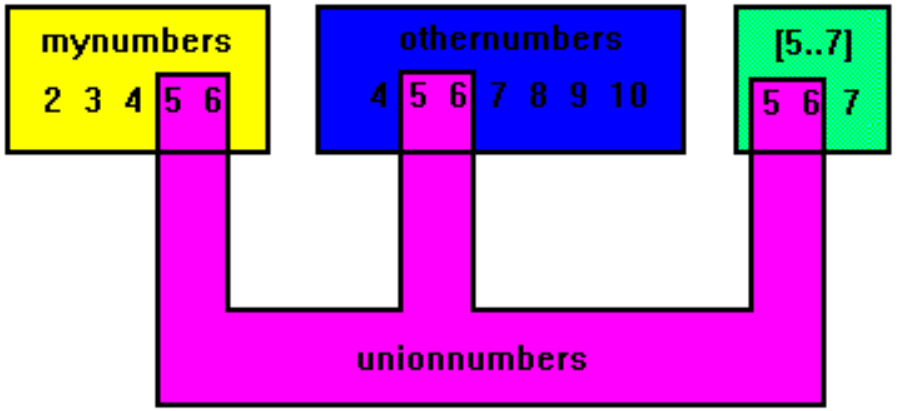
- Se A e B são conjuntos, então $A \cup B$ é o conjunto de elementos que são membros de A ou de B ou de ambos
- Se A e B são conjuntos, então $A \cap B$ é o conjunto de elementos que estão em A e em B
- Se A e B são conjuntos, então $A - B$ é o conjunto de elementos em A que não estão em B
- Exemplo: $A = \{a,b,c\}$ $B = \{b,d\}$
  $A \cup B = \{a,b,c,d\}$
  $A \cap B = \{b\}$
  $A - B = \{a,c\}$

unionnumbers: União, Diferença e Intersecção









Conjuntos em C

- Como implementar elegantemente um conjunto de inteiros de um intervalo predefinido em C?

- Vejam que este TAD é muito mais restrito do que um que manipula conjuntos em geral,
  - pois todos os conjuntos criados poderão ter, no máximo, de 0..TAM elementos. Assim, a união de 2 conjuntos terá, no máximo, TAM elementos.

Conjuntos em C

- Pode-se representar de forma eficiente via um vetor de booleanos (também conhecido por bitarray):

```
# define N 100 //por exemplo, conjunto que tem números de 0 a 99
int conjunto[N]; //conjunto[i]=1 se i está no conjunto; 0, caso contrário
```

Operações?

Operações usuais

- União(A,B,C)
- Intersecção(A,B,C)
- Diferença(A,B,C)
- Membro(x,A)
- Cria_Conj_Vazio(A)
- Insere(x,A)
- Remover(x,A)
- Atribui(A,B)
- Min(A)
- Max(A)
- Igual(A,B)
- Destroi_conjunto(A)
- Tamanho(A)
- Imprime_conjunto(A)



Definição das operações

- União(A,B,C): toma os argumentos A e B que são conjuntos e retorna A $\cup$ B à variavel C
- Intersecção(A,B,C): toma os argumentos A e B que são conjuntos e retorna A $\cap$ B à variavel C
- Diferença(A,B,C): toma os argumentos A e B que são conjuntos e retorna A - B à variavel C
- Membro(x,A): toma o conjunto A e o objeto x cujo tipo é o tipo do elemento de A e retorna um valor booleano – true se x $\in$ A e false caso contrário
- Cria_Conj_Vazio(A): faz o conjunto vazio ser o valor para a variável conjunto A. Primeira operação do conjunto.

Definição das operações

- Insere(x,A): toma o conjunto A e o objeto x cujo tipo é o tipo do elemento de A e faz x um membro de A. O novo valor de A = A $\cup$ {x}. Se x já é um membro de A, então a operação insere não muda A

- Remover(x,A): remove o objeto x, cujo tipo é o tipo do elemento de A, de A. O novo valor de A = A – {x}. Se x não pertence a A então a operação remove não altera A

- Destroi_Conj(A): remove o conjunto criado da memória.

- Imprime_conjunto(A): Imprime os elementos na forma de conjunto: { x, y, z }

Definição das operações

- Atribui(A,B): Seta o valor da variável conjunto A = ao valor da variável conjunto B
- Min(A): retorna o valor mínimo no conjunto A. Por exemplo: Min({2,3,1}) = 1 e Min({'a','b','c'}) = 'a'
- Max(A): Similar a Min(A) só que retorna o máximo do conjunto
- Igual(A,B): retorna true se e somente se os conjuntos A e B consistem dos mesmos elementos
- Tamanho(A): retorna o número de elementos do conjunto A

Exercício

- Em duplas, implementem em C o TAD conjunto de números inteiros com elementos do universo 0..TAM - 1
  - Use um arquivo .h

```
#define TAM $$$ // trabalha com elementos do universo 0..TAM -1

struct conjunto {
        int* v; //vetor booleano que armazenará o conjunto sendo que
                //o índice armazena o valor sendo true se o elemento está
                // no conjunto, false caso contrário
}
```