

# [ED248] TAD Conjunto (BooleanArrayIntSet)

Neste problema deverá apenas submeter uma classe **BooleanArrayIntSet** (e não um programa completo).

## O problema



A sua tarefa é criar uma classe **BooleanArrayIntSet**, que representa um conjunto de números inteiros, implementando o seguinte interface:

```
// Interface que define o TAD conjunto de numeros inteiros
interface IntSet {
    public boolean contains(int x); // Retorna true se x esta no conjunto
    public boolean add(int x); // Adiciona x ao conjunto (devolve true se conseguir)
    public boolean remove(int x); // Remove x do conjunto (devolve true se conseguir)
    public int size(); // Retorna o numero de elementos do conjunto
    public void clear(); // Limpa o conjunto (torna-o vazio)
    public boolean equals(IntSet s); // Retorna true se ambos os conjuntos sao iguais
    public IntSet intersection(IntSet s); // devolve um novo conjunto: a intersecao de ambos
}
```

Note que esta implementação deverá ser eficiente e demorar tempo constante nos métodos `contains`, `add` e `remove` (ou seja, uma implementação como a do **ArrayListIntSet** (descrita na aula prática) deverá ter exceder o tempo limite nos casos de teste grandes). A sugestão é usar um array de booleanos como explicado nas aulas e no guião.

Para testar pode fazer algo como o seguinte:

```
public class TestED248 {
    public static void main(String[] args) {
        int n = 1000000;
        IntSet s = new BooleanArrayIntSet(n); // Criar array de booleanos para numeros no intervalo [1,n]
        boolean tmp;

        System.out.println("Adicionando todos os numeros entre 1 e " + n + "...");
        for (int i=1; i<=n; i++)
            tmp = s.add(i);

        System.out.println("Adicionando novamente todos os numeros entre 1 e " + n + "...");
        for (int i=1; i<=n; i++)
            tmp = s.add(i);

        System.out.println("Verificando se todos os numeros entre 1 e " + n + " existem...");
        for (int i=1; i<=n; i++)
            tmp = s.contains(i);

        System.out.println("Removendo todos os numeros entre 1 e " + n + "...");
        for (int i=1; i<=n; i++)
            tmp = s.remove(i);
    }
}
```

Quando executado este código deverá demorar muito menos que 1 segundo e dar o seguinte output:

```
Adicionando todos os numeros entre 1 e 1000000...
Adicionando novamente todos os numeros entre 1 e 1000000...
Verificando se todos os numeros entre 1 e 1000000 existem...
Removendo todos os numeros entre 1 e 1000000...
```

## Input e Output

Deverá apenas submeter a classe **BooleanArrayIntSet**. O Mooshak irá criar várias instâncias da sua classe usando um construtor como mostrado no exemplo de utilização (**construtor com um argumento inteiro  $n$  indicando que deve suportar números no intervalo  $[1,n]$** ) e irá fazer uma série de testes aos métodos por si implementados.

É garantido que o conjunto nunca terá mais do que 100 números diferentes, e que todos os números serão inteiros positivos entre 1 e 1000 (inclusive) para todos os testes excepto o último. No último teste, destinado a verificar a eficiência, poderão ser usados números até 1 milhão e feitas milhões de adições, inserções ou verificações (**neste teste terá "Time Limit Exceeded" se a sua classe não for eficiente nestas operações**).

É garantido que os métodos são chamados de forma correcta (os argumentos fazem sentido e não geram excepções).

