

Velocidade de busca das listas e conjuntos

Crie a classe `TestaPerformance`, com um método `main` e um código que insere 50 mil números em uma `ArrayList` e os pesquisa. Vamos usar o método `currentTimeMillis()`, de `System`, para cronometrar o tempo gasto com a adição e pesquisa dos elementos:

```
public class TestaPerformance {  
  
    public static void main(String[] args) {  
  
        Collection<Integer> numeros = new ArrayList<Integer>();  
  
        long inicio = System.currentTimeMillis();  
  
        for (int i = 1; i <= 50000; i++) {  
            numeros.add(i);  
        }  
  
        for (Integer numero : numeros) {  
            numeros.contains(numero);  
        }  
  
        long fim = System.currentTimeMillis();  
  
        long tempoDeExecucao = fim - inicio;  
  
        System.out.println("Tempo gasto: " + tempoDeExecucao);  
  
    }  
  
}
```

Troque o `ArrayList` por `HashSet` e verifique o tempo que vai demorar:

```
Collection<Integer> numeros = new HashSet<>();
```

O que é lento? A inserção dos 50 mil elementos ou as 50 mil buscas? Descubra computando o tempo gasto em cada `for` separadamente.

Se você passar de 50 mil para um número maior, como 100 mil, verá que isso inviabiliza por completo o uso de uma `List` em casos que quisermos utilizá-la essencialmente para pesquisas.