

Sorting

1 Prerequisiti

- Conoscenza dei principali algoritmi di ordinamento

2 Introduzione e descrizione del Framework

Si vuole effettuare una comparazione delle performance tra gli algoritmi di ordinamento. Per questa ragione è stato predisposto un opportuno framework Java che permette l'implementazione e il test di algoritmi di ordinamento su array di interi.

2.1 Interfaccia Ordinamento

Interfaccia da implementare quando si vuole realizzare un nuovo algoritmo di ordinamento. I metodi da implementare sono:

```
public String getDescrizione()  
    descrizione dell'algoritmo (solo ai fini di reportistica)
```

Returns:

descrizione dell'algoritmo

```
public void sort(int[] array)
```

ordina l'array

Parameters:

array - array da ordinare

2.2 Classe BenchmarkSorting

La classe permette di effettuare un test congiunto di più algoritmi di ordinamento su un particolare input. Il metodo da utilizzare è il seguente:

```
public static void test(Ordinamento[] sorting,  
                        int size,  
                        TipoInput tipo)
```

effettua il test degli algoritmi sull'input indicato

Parameters:

sorting - algoritmi di ordinamento da testare

size - dimensione dell'array di input

tipo - tipo di input da generare (random ecc)

2.3 Enum TipoInput

```
public class TipoInput
```

Enumerazione che definisce i vari tipi di input che possono essere generati.

I valori ammessi sono:

- **RANDOM**,

- `ORDINATO_CRESCENTE`,
- `ORDINATO DECRESCENTE`,
- `COSTANTE`

2.4 Classe `OrdinamentoQuickSortDefaultJava`

Algoritmo di sorting (che implementa l'interfaccia `Ordinamento`) basato sul quicksort disponibile in java. Può essere usata come algoritmo di riferimento nei test.

2.5 Classe `TestOrdinamento`

Semplice classe di test che contiene un main di prova che effettua il test dell'algoritmo di ordinamento di default disponibile in Java.

3 Implementazione

Implementare le classi di test per i seguenti algoritmi di ordinamento:

- Quicksort
- Mergesort
- Bubblesort
- Selectionsort
- Insertionsort

Per ciascun algoritmo realizzare una opportuna classe che implementi l'interfaccia `Ordinamento`.

3.1 Test degli algoritmi

Modificando il metodo main della classe `TestOrdinamento`, effettuare il test comparativo degli algoritmi implementati andando ad agire su dimensione dei vettori e tipi di input.

4 Configurazione della JVM per la memoria

La JVM di default alloca 64MB di memoria per far girare i programmi. A volte questa quantità di memoria non è sufficiente ed è possibile aggiungere una opzione sulla linea di comando Java in modo da richiedere una differente memoria a disposizione. Per avviare un programma java con una specifica dimensione dello heap il comando è:

```
java -Xmx<memoria_in_MB>M <nome classe>
```

Esempio per allocare 512 MB il comando è:

```
java -Xmx512M <nome classe>
```