Laboratorio di Algoritmi e Strutture Dati
2015/16
Mini Progetto, Codice: MP3

Docente: Luca Tesei

Scadenza: 11 Maggio 2016 ore 23.59

Testo del mini progetto

- Si scriva il codice di due classi `QuickSort<E>` e `RandomizedQuickSort<E>` che implementino l'interface `SortingAlgorithm<E>` definita per scrivere algoritmi di ordinamento generici nel framework di valutazione numerica di algoritmi di ordinamento sviluppato a lezione (disponibile nel codice allegato). Le soluzioni devono implementare ricorsivamente l'algoritmo di quicksort e operare "in loco", cioè in ogni momento dell'esecuzione ci deve essere un numero costante di elementi della lista da ordinare che si trovano memorizzati fuori dalla lista data come input. La versione randomizzata del quicksort prevede che l'elemento pivot non sia sempre il primo elemento della sottosequenza che si deve ordinare, ma che sia scelto ogni volta randomicamente (utilizzare la classe `java.util.Random`) fra tutti gli elementi di detta sottosequenza. L'elemento scelto deve essere scambiato con quello in prima posizione.

- Si usi il framework di valutazione numerica degli algoritmi di ordinamento sviluppato a lezione per generare i dati relativi alle misurazioni di molteplici esecuzioni degli algoritmi scegliendo opportunamente le costanti che parametrizzano il framework[1]. Si usino poi tali dati per fare un'analisi comparata:

  - delle prestazioni dei due algoritmi sviluppati;
  - delle prestazioni degli stessi rispetto a tutti gli altri algoritmi disponibili nel codice allegato (sviluppati nelle lezioni precedenti).

  L'analisi dovrà confrontare le prestazioni rispetto al caso medio (la media delle misurazioni su input della stessa lunghezza), al caso ottimo (il massimo delle misurazioni su input della stessa lunghezza) e al caso pessimo

[1] Si veda l'interface `SortingAlgorithmEvaluationFrameworkParameters`.

(il minimo delle misurazioni su input della stessa lunghezza). Le misurazioni sono: il numero di confronti effettuati, i millisecondi impiegati per l'esecuzione e i nanosecondi impiegati per l'esecuzione.

Il framework genera due file .csv (comma separated value):

- `sequences.csv`, che contiente tutte le sequenze di numeri generate dal framework;
- `evalfram.csv`, che contiene le misurazioni relative a tutte le sequenze generate.

Tali file possono essere importati facilmente in qualsiasi foglio elettronico (ad esempio Excel o Open Office). Nel foglio elettronico poi i dati (principalmente di `evalfram.csv`) dovranno essere aggregati opportunamente e usati per generare i grafici necessari per la comparazione delle prestazioni. Un esempio (non completo) è dato tra i file allegati al progetto.

Le classi dovranno essere completamente autodocumentate tramite commenti interpretabili dall'utility `javadoc` e con commenti privati. Codice non adeguatamente commentato sarà valutato negativamente.

L'analisi dovrà essere presentata in forma scritta tramite una relazione. Alla relazione si dovrà allegare anche il foglio elettronico utilizzato per manipolare i dati grezzi ottenuti dal framework

Modalità di Consegna

I file `.java` per le classi, senza indicazione di package (cioè appartenenti al package di default), la relazione (in formato pdf) e i fogli elettronici usati per la relazione devono essere caricati entro la data di scadenza in una cartella Google Drive dal nome

`ASDL1516MP3-CognomeStudente-NomeStudente`

che deve essere condivisa, in sola lettura, tramite l'account

`nome-studente.cognome-studente@studenti.unicam.it`

con gli account:

- `luca.tesei@unicam.it` (docente) e
- `lorenzo.rossi@studenti.unicam.it` (tutor).

La cartella dovrà essere caricata in Google Drive come sotto-cartella della cartella `MP-CognomeStudente-NomeStudente` che verrà creata e condivisa in scrittura dal docente o dal tutor. Per la scadenza, farà fede la data dei file su Google Drive.

Allegati

- `sortingeval.zip` - Framework di valutazione e implementazione di Insertion Sort, Bubble Sort, Merge Sort (in due versioni) e Heap Sort.

- `provavalutazione.zip` - Esempio di importazione dei dati in Excel, esempio di aggregazione di dati (media delle prestazioni dei vari algoritmi) ed esempio di analisi comparata (grafico delle medie delle prestazioni dei vari algoritmi su dimensioni di input crescenti).