Corso di Algoritmi e Strutture Dati—Informatica per il Management

Primo Esercizio di Programmazione, 16/11/2010

Descrizione dell'esercizio

Si richiede di implementare una variante dell'algoritmo Quick Sort, che chiameremo QuickSort2(), come segue. Definiamo un parametro intero $k \geq 0$. L'algoritmo QuickSort2() deve ordinare un vettore di n numeri interi, in senso non decrescente, in due fasi. Nella prima fase si applica l'algoritmo Quick Sort come visto a lezione, con la differenza che la ricorsione termina quando la dimensione del sottovettore da ordinare diventa minore o uguale a k. Ovviamente, così facendo al termine della prima fase l'array in generale potrà non risultare ordinato. Per questo motivo, nella seconda fase si completa la procedura di ordinamento eseguendo l'algoritmo Bubble Sort sull'intero vettore.
Se k=0 oppure k=1, ovviamente QuickSort2() è esattamente uguale a Quick Sort, nel senso che al termine della prima fase l'array risulta già ordinato e la seconda fase risulta superflua. Se k>1, invece, la seconda fase risulta indispensabile per la correttezza dell'ordinamento.
Scopo di questa esercitazione è lo studio sperimentale del tempo di esecuzione dell'algoritmo QuickSort2() in funzione del parametro k. In particolare:

- Implementare in Java l'algoritmo QuickSort2() come descritto sopra (siete liberi di decidere come scegliere il pivot della fase di partizionamento); l'implementazione deve essere compilabile ed eseguibile sulle macchine Linux del laboratorio studenti (quindi devono essere conformi alla versione di Java installata);
- Scegliere una dimensione n del vettore da ordinare che sia sufficientemente grande per poter misurare i tempi di esecuzione di QuickSort2() (ad esempio $n=2^{18}$);
- Considerare diversi valori di k che siano potenze di due, partendo da 1. Ad esempio k=1, 2, 4, 8, 16... (è lasciata libertà di scegliere quanti diversi valori di k considerare, purché siano in numero congruo);
- Per ciascuno dei diversi valori di k, generare casualmente una serie (ad esempio 10) di vettori di n elementi, e misurare il tempo di esecuzione di QuickSort2() su tali vettori. Per ciascuna serie di prove calcolare il tempo medio di esecuzione di QuickSort2().
- Produrre un grafico in cui in ascissa ci siano i valori di k considerati, e in ordinata le medie dei tempi di esecuzione per ciascun set di esperimenti, come determinate nel punto precedente.
- (Facoltativo) Rispondere alla seguente domanda: quante iterazioni effettua (nel caso peggiore) l'algoritmo Bubble Sort nella seconda fase di QuickSort2()? Motivare brevemente la risposta.

Cosa consegnare

L'esercizio va svolto individualmente (non sono ammessi lavori di gruppo). La consegna va effettuata via mail, inviando un messaggio al docente (marzolla@cs.unibo.it) con titolo: "ASD2010 Progetto 1" **entro le ore 12:00 di lunedì 29 novembre 2010**. Riceverete un messaggio di conferma (non necessariamente immediato).
La mail deve provenire dal vostro account istituzionale (@cs.unibo.it, oppure @studio.unibo.it) e deve contenere il vostro Cognome, Nome e numero di matricola. Alla mail va allegato:

- un archivio <NomeCognome>.zip contenente i sorgenti Java del programma che implementa QuickSort2() e calcola i tempi medi. Si suggerisce di realizzare l'applicazione con un singolo file sorgente <NomeCognome>.java (vedere il template nella pagina web del corso); in questo caso è possibile allegare alla mail di consegna direttamente il file <NomeCognome>.java, senza bisogno di comprimerlo ulteriormente. **Il software consegnato deve essere compilabile ed eseguibile sulle macchine Linux del laboratorio studenti**.

- un documento PDF <NomeCognome>.pdf che contenga il grafico dei tempi medi di esecuzione al variare di k, e la risposta alla domanda di cui sopra (la risposta è facoltativa). Volendo è possibile (ma non obbligatorio) includere nel documento PDF un **BREVISSIMO** chiarimento sulle scelte implementative.

Quindi nel mio caso consegnerei i sorgenti nel file MorenoMarzolla.zip (oppure il singolo file sorgente MorenoMarzolla.java), e il documento MorenoMarzolla.pdf

Modalità di valutazione

Ciascun progetto verrà valutato un punto (che verrà sommato al voto finale della prova scritta, o alla media dei parziali) solo se tutte queste condizioni saranno soddisfatte:

- I sorgenti e il file PDF con grafico sono consegnati entro la scadenza.
- Il programma compila ed esegue correttamente sulle macchine Linux del laboratorio studenti; NOTA: non è indispensabile che i tempi di esecuzione riportati nel grafico siano misurati sulle macchine del laboratorio.
- L'implementazione è corretta.
- Tutto il materiale allegato è frutto del lavoro individuale di chi consegna.