

Relazione Laboratorio Algoritmi 2023-2024

Esercizio 1:

Obiettivo: Ordinare una serie di record caricati tramite un file .csv tramite 2 algoritmi di sorting. Tramite la funzione **clock** otteniamo i tempi di esecuzione dei metodi.

Mergesort

Scelte Implementative:

- **mergeSort()**: è una funzione **ricorsiva dicotomica**, il suo compito è dividere un array in sotto array in modo ricorsivo fino ad arrivare a elementi singoli.
- **merge()**: è la **funzione di tipo iterativo** che si occupa della ricostruzione dell'array e del sorting, partendo da elementi singoli fino alla ricostruzione completa e quindi alla dimensione originaria dell'array.

Prestazioni:

Campi	field1	field2	field3
10.000 tuple	0.006411	0.007466	0.007747
20milioni di tuple	9.863372 sec	9.616937 sec	10.074892 sec

Quicksort

Scelte Implementative:

- **quicksort()**: è una funzione **ricorsiva di tipo dicotomico** e si applica sulle due metà dell'array .
- **partition()**: è una **funzione di tipo iterativo** che ha lo scopo di trovare il pivot sul quale far agire quicksort() è di effettuare gli scambi tramite la funzione memcpv().

Prestazioni:

Campi	field1	field2	field3
10.000 tuple	0.007312	0.002631	0.005328
20 milioni di tuple	nv	nv	nv

Considerazioni finali:

Dato che il tempo del mergeSort è $O(n \log n)$, mentre quello del quicksort è sensibilmente superiore nel caso peggiore, mi aspetto che sia il più veloce dei due a concludere il sorting. Come osservato dai test il quicksort conclude l'esecuzione quando il carico di tuple è relativamente basso, nel caso dei 20 milioni di record il mergesort conclude sempre prima e il quicksort non conclude in tempi ragionevoli.

Esercizio 2:

Obiettivo: Contare la distanza di "edit" minima tra due parole (cancellazioni e inserimenti).

Edit distance

Scelte Implementative:

- **editDistance()**: è una funzione di tipo **ricorsivo** che si occupa di eseguire confronti tra due parole e calcolando quante cancellazioni e inserimenti sono eseguiti, infine restituisce il minore numero di operazioni eseguite. Se non vengono eseguite modifiche viene restituito il valore 0, non incrementando alcun contatore.

Edit distance dinamico

Scelte Implementative:

- **editDistanceDyn()**: è una funzione che crea una matrice inizializzando le celle al valore di -1, in seguito chiama editDistanceLoop().
- **editDistanceLoop()**: è una funzione ricorsiva che ha la stessa implementazione di editDistance con il supporto di matrice per migliorarne l'efficienza, salva i valori calcolati dai confronti.

Conclusioni:

Osservando i risultati utilizzando la matrice di supporto, il numero di confronti diminuisce in quanto vengono calcolati solo una volta, in questo modo vengono evitate operazioni già eseguite in precedenza e l'esecuzione conclude in un tempo minore.