

Analisi – Ordinamento funzioni

Ordinare le seguenti funzioni in accordo alla loro complessità asintotica. Si scriva $f(n) < g(n)$ se $O(f(n)) \subset O(g(n))$. Si scriva $f(n) = g(n)$ se $O(f(n)) = O(g(n))$, ovvero se $f(n) = \Theta(g(n))$.

$$f_1(n) = 2^{n+2}$$

$$f_2(n) = \log^2 n$$

$$f_3(n) = (\log_n(\sqrt{n})^2 n) + 1/n^2$$

$$f_4(n) = 3n^{0.5}$$

$$f_5(n) = 16^{n/4}$$

$$f_6(n) = 2\sqrt{n} + 4n^{1/4} + 8n^{1/8} + 16n^{1/16}$$

$$f_7(n) = \sqrt{(\log n)(\log n)}$$

$$f_8(n) = \frac{n^3}{(n+1)(n+3)}$$

$$f_9(n) = 2^n$$

Ricorrenza $2T(n/8) + 2T(n/4) + n$

Trovare un limite asintotico superiore e un limite asintotico inferiore alla seguente ricorrenza, facendo uso del metodo di sostituzione:

$$T(n) = \begin{cases} 1 & n \leq 1 \\ 2T(n/8) + 2T(n/4) + n & n > 1 \end{cases}$$

Trovare i limiti superiore e inferiori più stretti possibili per la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 2T(\lfloor n/2 \rfloor) + 4T(\lfloor n/4 \rfloor) + 15T(\lfloor n/8 \rfloor) + n^2 & n > 8 \\ 1 & n \leq 8 \end{cases}$$

Analisi – Algoritmo di selezione deterministico

Si consideri la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 11/5n + T(\lfloor n/5 \rfloor) + T(\lfloor 7n/10 \rfloor) & n > 1 \\ 1 & n \leq 1 \end{cases}$$

Individuare limiti inferiori e superiori tramite il metodo di sostituzione.

Analisi – MergeSortK

Si supponga di scrivere una variante di MergeSort, chiamata MergeSortK che, invece di suddividere l'array da ordinare in 2 parti (e ordinarle separatamente), lo suddivide in K parti, le ordina ognuna riapplicando MergeSortK, e le riunifica usando un'opportuna variante MergeK di Merge (la quale, naturalmente, fa la fusione su K sottoarray invece di 2). Come cambia, se cambia, la complessità temporale di MergeSortK rispetto a quella di MergeSort?

Cicli for

Si consideri il seguente pezzo di codice:

```
for  $i \leftarrow 1$  to  $n$  do  
   $\quad$  invoke( $i$ )
```

dove l'invocazione `invoke(i)` ha costo $O(i^h)$.

Si dimostri, per induzione su h , che $\sum_{i=1}^n i^h$ è $O(n^{h+1})$.

Esercizi di programmazione

Cerca la coppia - Versione 1

Dato un vettore $A[1 \dots n]$ di interi e un intero v , scrivere un algoritmo che determini se esistono due elementi in A la cui somma sia esattamente v .

Cerca la coppia - Versione 2

Dato un vettore $A[1 \dots n]$ di interi, scrivere un algoritmo che determini se esistono due elementi in A la cui somma sia esattamente 17.

Cerca la coppia - Versione 3

Dato un vettore $A[1 \dots n]$ di interi positivi, scrivere un algoritmo che determini se esistono due elementi in A la cui somma sia esattamente 17.