

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 $f(n) = \Theta(g(n))$.

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

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

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

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

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

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

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

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

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

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:

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

Ricorrenza

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:

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

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, lo suddivide in K parti, ri-ordina ognuna di esse applicando ricorsivamente MergeSortK, e le riunifica usando un'opportuna variante MergeK di Merge, che fonde K sottoarray invece di 2. Come cambia, se cambia, la complessità temporale di MergeSortK rispetto a quella di MergeSort?

SortinoSort

Il professor Sortino ha inventato un nuovo algoritmo di ordinamento.

- Il vettore di input viene diviso in tre parti, di dimensioni circa $n/3$.
- Vengono ordinati ricorsivamente i primi due terzi, i secondi due terzi, e infine di nuovo i primi due terzi.

```
SortinoSort(int[] A, int i, int j)
```

```
if  $j - i + 1 \leq 6$  then
```

```
    InsertionSort(A, i, j)
```

```
else
```

```
    int  $s = \lceil (j - i + 1) / 3 \rceil$ 
```

```
    SortinoSort(A, i, i + 2s - 1)
```

```
    SortinoSort(A, i + s, j)
```

```
    SortinoSort(A, i, i + 2s - 1)
```

- ① Qual è la complessità di questo algoritmo? Il Prof. Sortino finirà nella prossima edizione del mio libro?
- ② (Difficile, Opzionale) Dimostrare per induzione che questo algoritmo è corretto. Per comodità, assumete pure che tutti i valori siano distinti.

Trovare i limiti superiori e inferiori più stretti possibili per la seguente famiglia di equazioni di ricorrenza, per valori di a interi positivi.

$$T(n) = \begin{cases} aT(\lfloor n/2 \rfloor) + n^{a-1} & n \geq 2 \\ 1 & n < 2 \end{cases}$$