

Algoritmi e strutture dati

Allievi della Laurea Specialistica in Ingegneria Informatica

Tema d'esame - 30 Giugno 2003 – ore 9.00-10.30

NOME: **COGNOME:**

MATRICOLA: **FIRMA:**

Il valore degli esercizi è riportato nel prospetto a lato.

I diversi quesiti dello stesso esercizio sono reciprocamente indipendenti.

Esercizio	1	2	3	4a	4b	5	6a	6b
Valore	2	2	3	2	6	6	4	5
Valutazione								

Esercizi

- 1) Confrontare le prestazioni degli algoritmi di ordinamento noti.
- 2) Discutere analogie e differenze fra paradigma divide-et-impera e programmazione dinamica.
- 3) Introdurre il concetto di hashing universale.
- 4) Sia dato il problema di calcolare il valore b^n , dati i parametri di ingresso b e n , entrambi numeri (sicuramente) interi.
 - a) Scrivere in pseudocodice un algoritmo iterativo (il più semplice intuitivamente) che risolva il problema e svolgere l'analisi dei tempi di calcolo dello stesso.
 - b) Scrivere in pseudocodice un algoritmo ricorsivo, basato su una tecnica divide-et-impera, che risolva il problema e svolgere l'analisi dei tempi di calcolo dello stesso.
- 5) INSERTION-SORT può essere espresso come una procedura ricorsiva nel seguente modo: per ordinare $A[1..n]$, si ordina ricorsivamente $A[1..n - 1]$ e quindi si inserisce $A[n]$ nell'array ordinato $A[1..n - 1]$. Scrivere in pseudocodice questa versione ricorsiva di INSERTION-SORT e svolgere l'analisi del tempo di esecuzione della stessa.
- 6) La procedura BUILD-HEAP può essere realizzata usando ripetutamente HEAP-INSERT per inserire gli elementi nello heap. Si consideri la seguente realizzazione:

BUILD-HEAP'(A)

```
1 heap-size[A] ← 1
2 for i ← 2 to length[A]
3     do HEAP-INSERT(A, A[i])
```

- a) Le procedure BUILD-HEAP e BUILD-HEAP', se eseguite sullo stesso array di input, creano sempre lo stesso heap? Dimostrare che lo fanno o fornire un controesempio.
- b) Mostrare che, nel caso peggiore, BUILD-HEAP' richiede un tempo $\Theta(n/\lg n)$ per costruire uno heap di n elementi.