

## Seconda prova in itinere - 30 Giugno 2003

Cognome:

Nome:

Matricola:

Questa versione contiene cenni di una possibile soluzione.

Spazio riservato alla correzione

1	2	3	4	5	6	Tot.
/17	/17	/17	/16	/17	/16	/100

1. (17 punti) Si consideri il seguente frammento di programma.

```
LIST NewSort(LIST list)
{LIST SecondList;

if (list== NULL) return list;
else if (list->next == NULL) return list;
else {
    SecondList = split(list);
    return merge(OtherSort(list), OtherSort(SecondList));
}
}
```

Si supponga che *split* e *merge* siano le funzione studiate durante il corso, e che *OtherSort* sia una funzione che chiamata su una lista di  $m$  elementi restituisca la lista ordinata in tempo  $O(m^2)$ .

- a) La funzione *NewSort* é ricorsiva? **NO**  
 b) Si analizzi il tempo di esecuzione di *NewSort* su una una lista con  $n$  elementi,  $n$  pari.

**Il tempo di esecuzione é :**  $T(n) = \max\{O(1), O(1) + O(n) + O(n^2)\} = O(n^2)$ .

2. (17 punti) Si consideri la seguente funzione.

```
int nonsense(int n)
{int sum;

sum=0;
if (n==0) return 1234;
else if (n==1) return 2345;
else{
    for(i=0; i< n-1; i++)
        for (j=i+1; j< n; j++)
            sum = sum + 345*j;
    return nonsense(n-2);
}
}
```

- a) La funzione *nonsense* é ricorsiva? **SI**  
 b) Sia  $T(n)$  il tempo di esecuzione di *nonsense* su  $n$ . Si determini una relazione di ricorrenza per  $T(n)$ . (Nota: non e' richiesta la sua soluzione)

**La relazione di ricorrenza é:**  $T(0) = O(1), T(1) = O(1), T(n) = T(n-2) + O(n^2)$ , se  $n > 1$ .

3. (17 punti) Si risolva la seguente relazione di ricorrenza per  $T(n)$ :

$$T(0) = T(1) = 1$$

$$T(n) = T(n-2) + n/37.$$

**Una possibile soluzione:**  $T(n) = n/37 + (n-2)/37 + \dots + (n-2i)/37 + T(n-2i-2)/37$ . **Le iterazioni si fermeranno con  $n-2i-2=0$  oppure 1 secondo che  $n$  sia pari o dispari. In ogni caso  $T(n) \leq 1/37 \sum_{i=0}^n i = O(n^2)$ .**

4. (16 punti) Si consideri l'insieme  $S = \{1, 2, 3, 4, 5, 6, 7\}$ .

a) Qual è l'altezza minima di un albero binario di ricerca che contiene gli elementi di  $S$  e in che ordine possono essere inseriti gli elementi per ottenere tale altezza?

**L'altezza minima è 2 e si può ottenere inserendo in un albero binario inizialmente vuoto gli elementi di  $S$  nell'ordine: 4, 2, 6, 1, 3, 5, 7.**

b) Qual è l'altezza massima di un albero binario di ricerca che contiene gli elementi di  $S$  e in che ordine possono essere inseriti gli elementi per ottenere tale altezza?

**L'altezza massima è 6 e si può ottenere inserendo in un albero binario inizialmente vuoto gli elementi di  $S$  nell'ordine: 1, 2, 3, 4, 5, 6, 7.**

5. (17 punti) Si consideri la seguente funzione ricorsiva, con  $t$ , albero binario di ricerca, definito come durante il corso.

```
void inorder(TREE t)
{
  if (t != NULL){
    inorder(t->leftChild);
    printf('%c\n', t->nodeLabel);
    inorder(t->rightChild);
  }
}
```

Si mostri, tramite induzione strutturale, che la funzione *inorder* chiamata su un albero binario di ricerca  $t$ , stampa le etichette dei nodi dell'albero in ordine crescente.

**Bisogna definire un asserto  $S(t)$ , e provare il caso base (albero vuoto) e il passo induttivo (albero con radice avente un sotto-albero sinistro e uno destro) utilizzando la proprietà dell'ordinamento in un albero binario di ricerca.**

6. (17 punti) Scrivere in C una funzione ricorsiva che, presa una lista di interi compresi fra 18 e 30, restituisca il numero di elementi maggiori o uguali a 24.

**Bisogna scegliere una struttura di dati per la lista e in tale struttura definire una funzione che nel caso base (lista vuota) restituisse zero; negli altri casi confrontasse il primo elemento  $x$  con 24 e restituisse il valore della funzione stessa sulla lista dal secondo elemento in poi, se  $x < 24$ , ovvero tale valore incrementato di 1, altrimenti.**