

Fondamenti di Informatica A-L – (Prof. P. Zingaretti)

PROVA SCRITTA – 20 giugno 2005

Avvertenze:

- * Consegnare **solo fogli formato A4 scritti su un solo lato** (no fronte-retro)
- * **In testa a ciascun foglio** scrivere: cognome, nome, numero progressivo di pagina rispetto al totale
esempio per il secondo foglio di 3 consegnati: Giuseppe Russo 2/3
- * Mantenere sul banco il **libretto o altro documento di riconoscimento** fino a controllo avvenuto
- * Nient'altro deve trovarsi sul banco: **non è consentito consultare libri, dispense, appunti, ecc.**
- * La **correzione** di riferimento per l'autovalutazione verrà effettuata in questa stessa aula alle ore **13:45**
- * La consegna delle **fotocopie** dei compiti avverrà al termine della correzione
- * La **prova orale** si terrà **lunedì 27 giugno** alle ore **9:00** presso il D.I.I.G.A.



Si ricorda che chi si presenterà all'orale **DEVE** portare l'implementazione al computer della propria soluzione, eventualmente corretta, **corredata di tutto quanto necessario alla verifica** del corretto funzionamento.

1. (10 punti)

Si definisca una funzione C che riceve in input un array di N (variabile) valori interi tutti diversi tra loro e restituisce il valore dell'elemento mediano (cioè, l'elemento che ha tanti maggioranti quanti minoranti se N è dispari, oppure un maggiorante in più se N è pari).

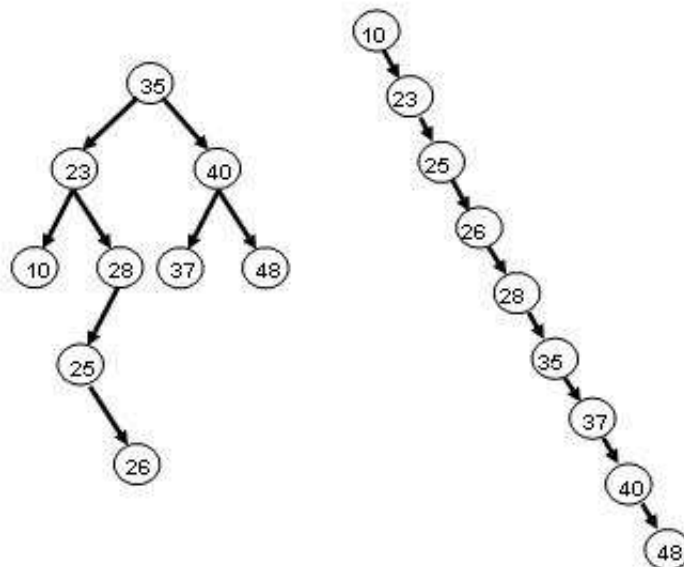
Esempi: con $A=\{2, 7, 15, 4, 23\}$ viene restituito 7; con $A=\{2, 7, 15, 4\}$ viene restituito 4

2. (20 punti)

Si scriva una funzione C che riceve in input un albero binario di ricerca in cui a ciascun nodo è associato un valore intero memorizzato nel campo `val`.

La funzione restituisca in output il nuovo (cioè quello originale non deve essere modificato) albero binario di ricerca con lo sbilanciamento massimo; cioè con soli discendenti destri.

Esempio:



1. (10 punti)

Si definisca una funzione C che riceve in input un array di N (variabile) valori interi tutti diversi tra loro e restituisce il valore dell'elemento mediano (cioè, l'elemento che ha tanti maggioranti quanti minoranti se N è dispari, oppure un maggiorante in più se N è pari).

Esempi: con A={2, 7, 15, 4, 23} viene restituito 7; con A={2, 7, 15, 4} viene restituito 4

```
int mediano(int *v, int N);
void bubbleSort (int *v, int N);

void bubbleSort(int v[], int N){ //6 punti
    int i, ordinato=0;
    while (N>1 && !ordinato){
        ordinato=1;
        for (i=0; i<N-1; i++){
            if (v[i]>v[i+1]) {
                int tmp=v[i]; v[i]=v[i+1]; v[i+1]=tmp;
                ordinato=0;
            } //end if
        }
        N--;
    } //end while
}

int mediano (int *v, int N) {
    int index=N/2 - ((N%2)? 0 : 1); //4 punti
    bubbleSort(v, N);
    return v[index];
}
```

2. (20 punti)

Si scriva una funzione C che riceve in input un albero binario di ricerca in cui a ciascun nodo è associato un valore intero memorizzato nel campo val.

La funzione restituisca in output il nuovo (cioè quello originale non deve essere modificato) albero binario di ricerca con lo sbilanciamento massimo; cioè con soli discendenti destri.

```
typedef struct btree_tag { int value; struct btree_tag *left;
                          struct btree_tag *right; } btree;

void sbilancia (btree **new_root, btree *btp) {
    if (btp!=NULL) {
        sbilancia(new_root, btp->left);
        if (*new_root) { //radice già trovata! (4 punti)
            (*new_root)->right=(btree *)malloc(sizeof(btree));
            *new_root=(*new_root)->right;
        }
        else (*new_root)=(btree *)malloc(sizeof(btree)); //init
        (*new_root)->value=btp->value;
        (*new_root)->left=NULL;
        (*new_root)->right=NULL;
        sbilancia(new_root, btp->right);
    }
}
```