

## Algoritmica e Laboratorio A

### Esercitazione del 03/5/2013

#### Esercizio: ISAVL?

Dato un albero binario ricerca di puntatore **u** definire un algoritmo dica che l'albero è o meno un AVL.

Soluzione:

Si risolve con una Post-Visita, ove ogni sotto-albero restituisce a termine visita i parametri <bool, alt>, il primo che dice se il sottoalbero è AVL il secondo l'altezza del sottoalbero>

```
ISAVL?(u):
IF (u == Null) RETURN <TRUE,-1>
ELSE {
    <ISsx, altsx> = ISAVL?(u.sx);
    <ISdx, altdx> = ISAVL?(u.dx);
    alt = max(altsx, altdx)+1;
    IS = ISsx&&ISdx&&( |altsx- altdx| <=1)
    RETURN <IS, alt>
}
```

Complessità  $T(n) = O(n)$ .

#### Esercizio: ISABR?

Dato un albero binario stabilire se è un albero binario di ricerca.

Soluzione:

Due versioni. La prima usa una Post-Visita e di ogni sottoalbero restituisce 3 valori <bool, min, max>, rispettivamente il fatto che il sottoalbero sia o meno ABR, il minimo e il massimo del sottoalbero.

Ricorsivamente un nodo radice di sottoalbero deve essere minore del max del sottoalbero sinistro (predecessore) e maggiore del min del sottoalbero destro (successore).

```
ISABR1?(u):
IF (u == Null) RETURN <TRUE,-∞,∞>
ELSE {
    <ISsx, minsx, maxsx>= ISABR1?(u.sx);
    <ISdx, mindx, maxdx>= ISABR1?(u.dx);
    IF (ISsx&&ISdx&&( maxsx< u.dato.chiave < mindx)){
        IS=TRUE;
        min= minsx;
        max= maxdx;
    }
    ELSE RETURN <FALSE,-,->
    RETURN <TRUE,min,max>
}
```

Complessità  $T(n) = O(n)$ .

Una versione alternativa si ottiene con una variante della Visita Simmetrica, dove ad ogni passo, si confronta il nodo con l'ultimo elemento visitato (parametro last). Il parametro deve essere sia d'ingresso che di uscita in quanto deve propagare sia informazioni che provengono dagli antenati che dai discendenti. I parametri di uscita sono ora <IS, last>

```
ISABR2(u, last):
IF(u == Null) RETURN < TRUE, 0 >;
ELSE {
  IF (u.sx != null) {
    <ISsx, lastsx> = ISABR2?(u.sx, last);
    IF (ISsx == FALSE) RETURN < FALSE, 0>;
    ELSE last = lastsx;
  }
  IF(last > u.dato.chiave) RETURN <FALSE, 0>;
  IF (u.dx == null) RETURN < TRUE, u.dato.chiave >;
  ELSE return ISABR2?(u.dx, u.dato.chiave);
}
```

Complessità  $T(n) = O(n)$ .

#### **Esercizio: Costruzione albero AVL**

Sia data la sequenza di chiavi  $S = \{15, 9, 30, 8, 10, 20, 45, 2, 37, 7, 12, 3, 36\}$ .

Inserirle in un albero AVL inizialmente vuoto, indicando a ogni inserimento l'eventuale nodo critico e l'operazione di ribilanciamento eseguita.

Eliminare poi le chiavi 20 e 30.

#### **Esercizio Successore**

Progettare un algoritmo per realizzare l'operazione SUCCESSORE in un albero binario di ricerca.

#### **Esercizio Calcolo dell'altezza nell'immagine binaria**

Progettare un algoritmo che calcoli il numero di foglie di un albero ordinale rappresentato in forma binarizzata.