

mercoledì 14 marzo 2012
III compitino

Teoria) 1) Scrivere PRE e POST opportune per la seguente funzione ricorsiva:

//PRE= ??

```
bool H(nodo*L, int & k)
{ if(!L)    { k=1; return false; }
  if(H(L->next, k)) {k=k+1; return false;}
  else return true;
}
```

POST=

lung pari => false e $k = \text{lung}/2 + 1$

lung dispari => true e $k = (\text{lung}-1)/2 + 1$

PRE=(r punta ad un albero binario corretto (event. vuoto), $k \geq 0$ e y intero qualsiasi)
POST=(F restituisce un valore booleano v tale che v sse nell'albero puntato da r c'è un cammino che contenga esattamente k nodi (non necessariamente contigui) con campo info=y).

Programmare) Scrivere una funzione ricorsiva con il seguente prototipo: `bool F(nodo* r, int k, int y);`
e che soddisfa la seguente coppia di PRE e POST condizioni:

PRE=(r punta ad un albero binario corretto (event. vuoto), $k \geq 0$ e y intero qualsiasi)
POST=(F restituisce un valore booleano v tale che v sse nell'albero puntato da r c'è un cammino **radice-foglia** che contenga esattamente k nodi (non necessariamente contigui) con campo info=y).

Il tipo nodo è `struct nodo{int info; nodo*left,*right;};`
Si richiede di dare una dimostrazione induttiva della correttezza della funzione F rispetto alla PRE e POST date.

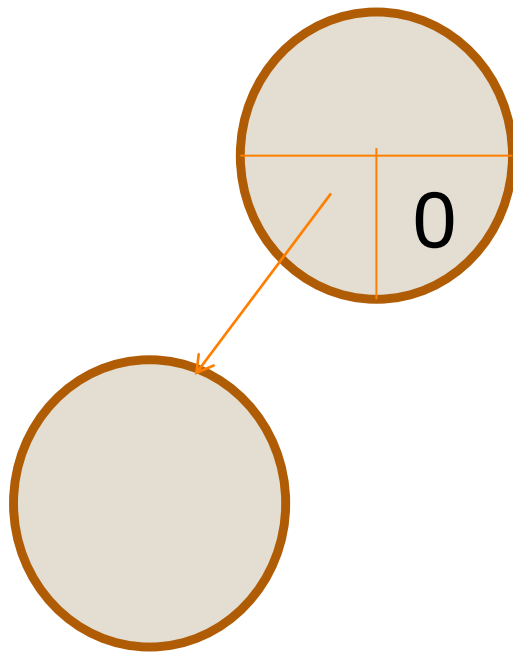
caso base:

- albero vuoto ?? non ha nodi e quindi non ha cammini e quindi
- la risposta DEVE essere false
- ricorsione si ferma alle foglie

sembrano in contrasto: se ci fermiamo sulle foglie che senso ha considerare l'albero vuoto?

Solo per la prima invocazione? NO

ci sono nodi con 1 solo figlio



qui troviamo un albero vuoto
e la risposta false è
necessaria altrimenti
considereremmo anche
cammini non radice foglia

usiamo

```
bool leaf(nodo*r)
{return !r->right && !r->left;}
```

```
bool ex(nodo* r,int k, int y)
{
    if(!r)
        return false;

    if(leaf(r))
        if((r->info==y && k==1)|| (r->info !=y &&
            k=0))
            return true;
        else
            return false;
    else
```

.....

```
else // r non e' foglia, passo ricorsivo
{
    if(r->info==y)
        k--;

    return ex(r->left,k,y)||ex(r->right,k,y);
}
```

manca ulteriore caso base:

```
if(k<0)
    return false;
```


dimostrazione:

casi base:

r vuoto \Rightarrow false \Rightarrow POST

r =foglia c'è cammino sse

- $k = 0$ e $r \rightarrow \text{info} \neq y$

- $k = 1$ e $r \rightarrow \text{info} == y$

il programma restituisce true solo in questi casi e altrimenti ritorna false
 \Rightarrow POST

ulteriore caso base:

$r \rightarrow \text{info} = y$ e $k = 0$

il programma restituisce **false**

vale la POST?

si tratta di dimostrare che non ci può
essere cammino con 0 y in r

visto che la radice ha y è ovvio che sia così
 \Rightarrow POST

passo ricorsivo:

se $r \rightarrow \text{info} = y$, $k = vK - 1$, allora c'è cammino con v_k y in r sse c'è cammino con k r nel sottoalbero sinistro o destro
è esattamente quello che ritorna la funzione

se $r \rightarrow \text{info} \neq y$, c'è cammino con k y in r sse c'è cammino con k r nel sottoalbero sinistro o destro
è esattamente quello che ritorna la funzione