

**CORSO DI
ALGORITMI E STRUTTURE DATI**

Prof. ROBERTO PIETRANTUONO

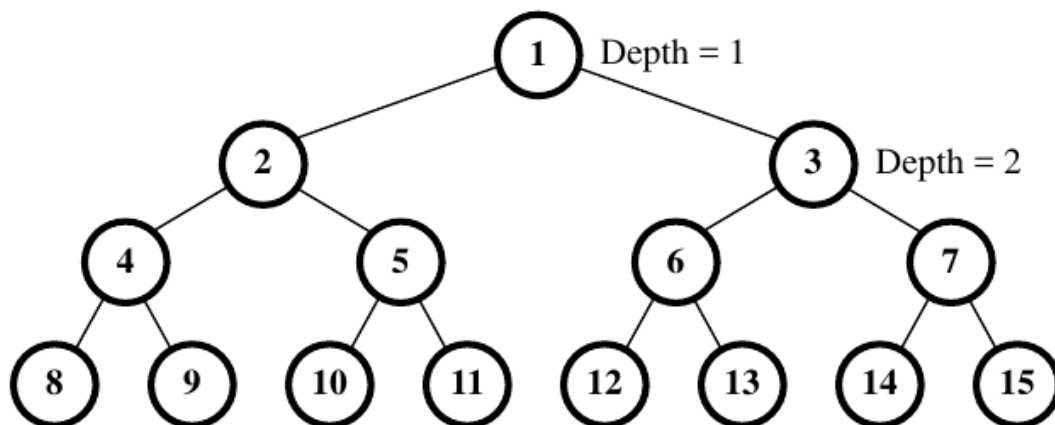
Prima prova - 16/02/2022

Indicazioni

Si consegnino un file in **formato editabile (.txt, .docx, .rtf, etc.)** nominandolo "*CognomeNome*", in cui è riportata l'implementazione (nel linguaggio scelto) seguita da una indicazione della complessità temporale dell'algoritmo implementato (complessità nel caso peggiore, è sufficiente il limite superiore $O(f(n))$). Se si utilizzano librerie di cui non si conosce la complessità, lo si indichi nella spiegazione (ad esempio, "la complessità è $O(n \log n)$ al netto della complessità dell'algoritmo x , che è non nota"). Se si utilizza la randomizzazione, si indichi anche il tempo di esecuzione atteso.

PROBLEMA 1

Si consideri un albero binario completamente pieno, con i nodi numerati come rappresentato in figura (ossia numeri crescenti procedendo dall'alto al basso e da sinistra a destra). Un certo numero di palline vengono lasciate cadere una alla volta dalla radice finché giungono ad un nodo foglio. Ad ogni nodo non terminale, la pallina va a sinistra se trova il valore del flag X associato al nodo non terminale pari a falso ($X = F$), oppure a destra se trova tale flag settato a vero ($X = V$). Inizialmente tutti flag sono falsi; quando una pallina visita un nodo non terminale cambia il valore corrente del flag, settandolo a VERO se era FALSO (proseguendo poi la sua discesa verso sinistra), ed a FALSO se invece era VERO (proseguendo verso destra). Quindi, la prima pallina cadrà in posizione 8, la seconda in posizione 12, la terza in posizione 10 e così via.



Si scriva un programma per determinare la posizione finale di arresto P per ogni test case. Per ogni test case l'intervallo di due parametri D e I è il seguente:

Si consideri una serie di casi di test, per ciascuno dei quali sono forniti due valori interi: un valore $Depth$, che è la profondità massima dell'albero (ad es. $Depth = 4$ in figura), ed un secondo valore N che indica l' N -esima palla che viene lasciata cadere. (si assume anche N non superi il numero di nodi foglia ($2^{(Depth-1)}$)). Si scriva un programma per determinare la posizione finale della N -esima palla nell'albero di profondità $Depth$. (si assumo $2 \leq Depth \leq 10$)

INPUT

La prima linea dell'input indica il numero di casi di test T . A seguire, ogni linea indica la coppia Depth N .

OUTPUT

Ogni linea dovrà riportare la posizione finale della pallina

Sample Input

```
5
4 2
2 2
8 128
3 4
10 1
```

Sample Output

```
12
3
255
7
512
```

PROBLEMA 2

Dati due interi x e n , si scriva una funzione che calcola x^n con complessità $O(\log n)$.
Si assuma di non dover gestire l'overflow.

INPUT

Ogni linea contiene un caso di test, in cui il primo numero è x ed il secondo è n

OUTPUT

Sample Input

```
2 3
2 5
7 2
```

Sample Output

```
8
32
49
```