

## Algoritmi e Strutture Dati – 16/06/2021 – Parte A

### A1 – Complessità – Punti $\geq 8$

Si consideri la seguente funzione di complessità:

$$T(n) = \begin{cases} 16T(n/4) + f(n) \end{cases}$$

Calcolare la complessità di  $T(n)$  per i seguenti valori di  $f(n)$ , utilizzando il Master Theorem (normale o esteso); scrivere "N/A" (non applicabile) se non è possibile utilizzare nessuna delle due versioni; nel caso si utilizzi il Caso 3 del Master Theorem versione estesa, si dimostri anche la condizione aggiuntiva.

1.  $f(n) = n$
2.  $f(n) = n \log n$
3.  $f(n) = n^2$
4.  $f(n) = n^2 + \log n$
5.  $f(n) = n^2 / \log n$
6.  $f(n) = n^2 \log n$
7.  $f(n) = n^3$
8.  $f(n) = n^3 \log n$

### A2 – Trova la coppia – Punti $\geq 10$

Scrivere un algoritmo

**boolean** findPair(**int**[]  $A_1$ , **int**[]  $A_2$ , **int**  $n$ , **int**  $k$ )

che prenda in input due vettori ordinati di interi contenenti  $n$  elementi ciascuno e un intero  $k$ , e restituisca **true** se esistono due valori  $a_1 \in A_1$  e  $a_2 \in A_2$  tali che  $a_1 + a_2 = k$ , **false** altrimenti.

Discutere informalmente la correttezza dell'algoritmo e calcolare la sua complessità computazionale.

Per esempio, se  $A_1 = [1, 5, 8, 10]$ ,  $A_2 = [31, 37, 43, 52]$  e  $k = 51$ , l'algoritmo deve restituire **true** perché esiste la coppia  $8 + 43$ ; se invece  $k = 6$ , l'algoritmo deve restituire **false** (in quanto  $1 + 5$  non possono essere considerati in quanto appartengono allo stesso vettore).

### A2 – isComplete – Punti $\geq 12$

Un albero binario *completo* è un albero binario tale che:

- Tutte le foglie hanno profondità  $h$  o  $h - 1$
- Tutti i nodi a livello  $h$  sono "accatastati" a sinistra
- Tutti i nodi interni hanno grado 2, eccetto al più uno

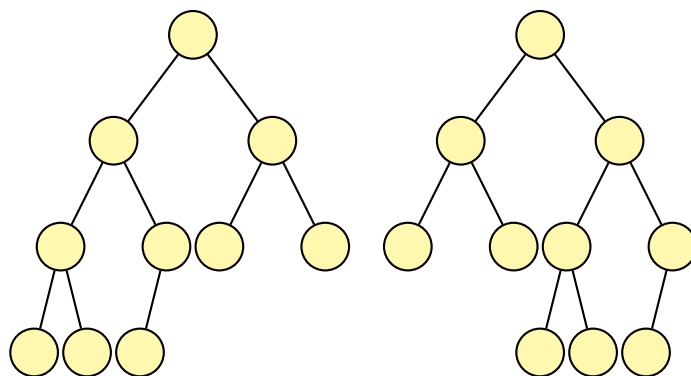
Scrivere un algoritmo

**boolean** isComplete(**TREE**  $T$ )

che prenda in input un albero binario  $T$  e restituisca **true** se l'albero è completo, **false** altrimenti.

Discutere informalmente la correttezza dell'algoritmo e calcolare la sua complessità computazionale.

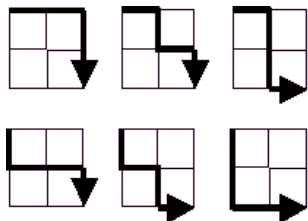
Dei due alberi seguenti, quello di sinistra è completo, quello di destra no. Vi ricordo che abbiamo introdotto gli alberi binari completi parlando della struttura dati *heap*.



## Algoritmi e Strutture Dati – 16/06/2021 – Parte B

### B1 – Stampa percorsi – Punti $\geq 8$

Partendo dall'angolo in alto a sinistra di una griglia  $2 \times 2$ , e muovendosi solo a destra o in basso, ci sono esattamente 6 percorsi distinti per raggiungere il bordo in basso a destra.



Questi percorsi possono essere rappresentati dalle stringhe RRDD, RDRD, RDDR, DRRD, DRDR, DDRR, dove D sta per down, R sta per right.

Scrivere un algoritmo

```
printPaths(int n)
```

che prenda in input un intero positivo  $n$  e stampi tutti i percorsi distinti per andare dall'angolo in alto a sinistra all'angolo in basso a destra di una griglia  $n \times n$ , muovendosi solo a destra e in basso.

Discutere correttezza e complessità dell'algoritmo proposto.

### B2 – removeSum – Punti $\geq 10$

Scrivere un algoritmo

```
int removeSum(int[] A, int n, int C)
```

che prenda in input un vettore  $A$  contenente  $n$  interi distinti e un valore positivo  $C$ , e restituisca la dimensione del più grande insieme di valori che può essere rimosso da  $A$  tale per cui la somma dei valori rimanenti è pari a  $C$ .

Se non è possibile ottenere  $C$  l'algoritmo deve restituire  $-\infty$ ; oppure potete assumere che sia sempre possibile ottenere  $C$  in almeno un modo. Scegliete l'approccio che preferite.

Discutere correttezza e complessità dell'algoritmo proposto.

Per esempio, se  $A = [5, 1, 7, 2, 3, 9, 8]$  e  $C = 17$ , è possibile ottenere  $C$  rimuovendo  $\{1, 8, 9\}$ ,  $\{2, 7, 9\}$ ,  $\{1, 3, 5, 9\}$ ,  $\{2, 3, 5, 8\}$ ,  $\{1, 2, 3, 5, 7\}$ . L'ultimo caso è quello che presenta il maggior numero di termini e quindi l'algoritmo dovrà restituire 5.

### B3 – Massima somma crescente – Punti $\geq 12$

Scrivere un algoritmo

```
int maxSumIncreasing(int[] A, int n)
```

che prenda in input un insieme di interi positivi  $A$  e restituisca il valore della sottosequenza crescente di somma massima.

Discutere correttezza e complessità dell'algoritmo proposto.

Per esempio, nel vettore  $[2, 102, 3, 4, 101, 5, 6]$ , la sottosequenza crescente di somma massima è  $2 + 3 + 4 + 101$ , il cui valore è 110.