Cognome: **Nome**: **# Matricola**: **Riga**: **Col**:

Algoritmi e Strutture Dati – Parte A – 18/01/2021

Esercizio A1 – Punti ≥ 8

Trovare i limiti superiore e inferiore più stretti possibili per la seguente equazione di ricorrenza:

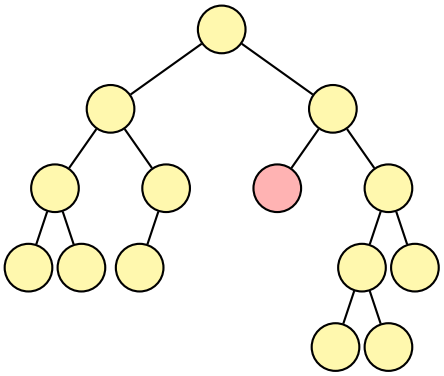
$$T(n) = \begin{cases} 1 & n < 8 \\ 7T(\lfloor n/2 \rfloor) + 4T(\lfloor n/4 \rfloor) + n^3 \log n & n \geq 8 \end{cases}$$

Esercizio A2 – Profondità minima – Punti ≥ 10

Scrivere un algoritmo che prenda in input un albero binario T e restituisca la sua profondità minima, ovvero la minima profondità fra tutte le profondità delle foglie.

Spiegare il funzionamento e discutere la complessità dell'algoritmo proposto.

Ad esempio, nell'albero sottostante, la profondità minima è pari a 2, la profondità del nodo rosa.



Esercizio A3 – Individua il singolo – Punti ≥ 12

Sia A un vettore contenente $n \geq 1$ interi, *non necessariamente ordinato*, con n dispari. Nel vettore sono memorizzati un certo numero di interi distinti; a parte un valore che compare una volta sola, tutti gli altri interi compaiono due volte in posizioni consecutive. Scrivere un algoritmo

$$\textbf{int}\ \mathsf{findSingle}(\textbf{int}[\,]\ A,\ \textbf{int}\ n)$$

che restituisca l'intero che compare una volta sola.

Spiegare il funzionamento e discutere la complessità computazionale dell'algoritmo proposto. *Soluzioni in tempo lineare o superiore non verranno considerate.*

Ad esempio, con l'input

$$A = [1, 1, 4, 4, 2, 2, 0, 1000, 1000, -2, -2]$$

l'algoritmo deve restituire 0 in quanto l'unico numero che compare una volta sola.

Cognome: **Nome**: **# Matricola**: **Riga**: **Col**:

Algoritmi e Strutture Dati – Parte B - 18/01/2021

Esercizio B1 – Parentesizzazioni – Punti ≥ 8

Scrivere una funzione

printPar(**int** n)

che stampi tutte le possibili parentesizzazioni corrette composte da n coppie di parentesi.

Spiegare il funzionamento e discutere la complessità dell'algoritmo proposto.

Ad esempio, con $n = 3$, le possibili parentesizzazioni corrette sono:

```
()()()      (()())
(())()      ()(())
((()))
```

Queste non sono parentesizzazioni corrette con $n = 3$:

```
)))(((      )()()(
))))))      ()()
```

Esercizio B2 - Stiamo compatti - Punti ≥ 10

Scrivere un algoritmo

int minRemove(**int**[] A, **int** n, **int** k)

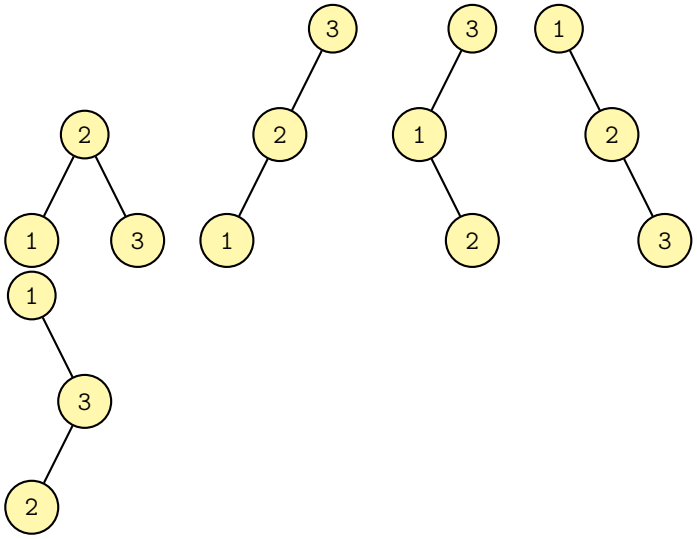
che prenda in input un vettore A contenente n interi distinti e un valore positivo, e restituisca il più piccolo numero di elementi da rimuovere da A, in modo che l'insieme S dei valori rimanenti è tale per cui $\max(S) - \min(S) \leq k$.

Spiegare il funzionamento e discutere la complessità dell'algoritmo proposto.

- $A = [-3, 3, 4, 7, 9, 12, 15]$ e $k = 6$: la risposta è 3, perché rimuovedendo $-3, 12, 15$ si ottiene il sottoinsieme $S = \{3, 4, 7, 9\}$ per cui $\max(S) - \min(S) = 9 - 3 \leq k = 6$ e non esistono sottoinsiemi di rimozione più piccoli che rispettano la condizione.
- $A = [-30, 4, 1, -40, 3, 2, -2, -20]$ e $k = 4$: la risposta è 4, perché rimuovendo $-30, -40, -2, -20$ si ottiene il sottoinsieme $S = \{4, 1, 3, 2\}$ per cui $\max(S) - \min(S) = 4 - 1 = 3 \leq k = 4$, e non esistono sottoinsiemi di rimozione più piccoli che rispettino la condizione.

Esercizio B3 - Stessa visita - Punti ≥ 12

Se eseguite una visita in profondità simmetrica su tutti questi alberi, la sequenza di visita è sempre la stessa: 1, 2, 3.



Scrivere un algoritmo

int sameVisit(**int** n)

che prenda in input un intero positivo o nullo n e restituisca il numero totale di alberi strutturalmente diversi, contenenti n nodi numerati da 1 a n la cui visita in profondità simmetrica dia origine alla sequenza 1, 2, $\ldots n$.

Spiegare il funzionamento e discutere la complessità dell'algoritmo proposto.

Ad esempio, per $n = 3$ l'algoritmo deve restituire 5, in quanto gli alberi in figura sono tutti e soli gli alberi strutturalmente diversi contenenti 3 nodi la cui visita dà origine a 1, 2, 3.