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

Algoritmi e Strutture Dati – Parte A – 03/07/2020

Esercizio A1, punti ≥ 8

Trovare i limiti superiori e inferiori più stretti possibili per la seguente equazione di ricorrenza, utilizzando il metodo di sostituzione (detto anche per tentativi):

$$T(n) = \begin{cases} 3T(\lfloor n/3 \rfloor) + 4T(\lfloor n/4 \rfloor) + \\ 5T(\lfloor n/5 \rfloor) + 6T(\lfloor n/6 \rfloor) + n^2 & n \geq 6 \\ 1 & n < 6 \end{cases}$$

Esercizio A2, punti ≥ 11

Sia T un albero binario contenente $n \geq 1$ nodi e A un vettore *ordinato* contenente n interi distinti. Oltre ai normali campi di un albero binario, ogni nodo t di T contiene un campo $t.size$ che contiene il numero di nodi del sottoalbero radicato in t (già inizializzato) e un campo $t.value$ (inizialmente vuoto). Ovviamente, $T.size = n$.

Scrivere un algoritmo

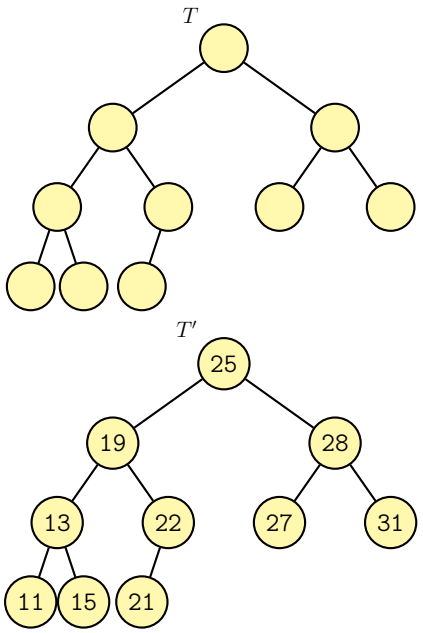
$$\mathsf{binaryInsert}(\textsc{Tree}\ T,\ \textbf{int}[\,]\ A,\ \textbf{int}\ n)$$

che inserisca tutti i valori di A in T, in modo tale che l'albero risultante sia un albero binario di ricerca.

Discutere correttezza e complessità computazionale dell'algoritmo proposto.

Esempio:

- l'albero T dato in input (senza valori);
- l'albero T' dopo che sono stati inseriti i valori $A = [11, 13, 15, 19, 21, 22, 25, 27, 28, 31]$.



Esercizio A3 – Maggioranza – Punti ≥ 11

Scrivere una funzione

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

che prenda in input un vettore *ordinato* A contenente n interi e restituisca **true** se A contiene un valore di maggioranza, ovvero un valore che compare più di $n/2$ volte; restituisca **false** altrimenti.

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

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

Algoritmi e Strutture Dati – Parte B - 03/07/2020

Esercizio B1, punti ≥ 8

Si scriva un algoritmo

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

(largest palindrome permutation) che prenda in input un numero arbitrariamente grande, rappresentato da n cifre decimali comprese fra 0 e 9 memorizzate nel vettore A, e restituisca (nello stesso vettore) una permutazione delle cifre che sia palindroma e rappresenti il numero più alto possibile fra tutte le permutazioni palindrome. Ad esempio, per $A = [1, 1, 3]$ si deve restituire $A = [1, 3, 1]$, interpretato come 131; per $A = [2, 5, 1, 5, 1]$, si deve restituire $A = [5, 1, 2, 1, 5]$, in quanto $51215 > 15251$ (l'unica altra permutazione palindroma)

Discutere correttezza e complessità dell'algoritmo proposto. Per semplicità, assumete che sia sempre possibile ottenere permutazioni palindrome.

Esercizio B2, punti ≥ 10

Scrivere un algoritmo

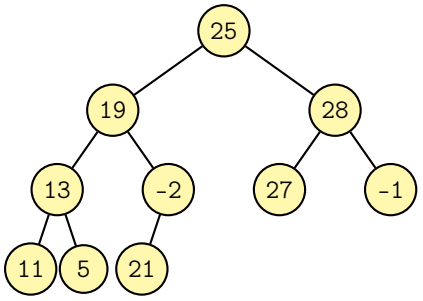
$$\mathsf{printPositive}(\textsc{Tree}\ T)$$

che prenda in input un albero binario T non vuoto, con valori contenuti nel campo *value* e stampi i valori di tutti i percorsi radice-foglia composti da solo valori positivi.

Discutere correttezza e complessità computazionale dell'algoritmo proposto.

Ad esempio, nell'esempio, i percorsi da stampare sono:

```
25 19 13 11
25 19 13 5
25 28 27
```



Esercizio B3, punti ≥ 12

Una sequenza k-limitata è una sequenza di valori tali per cui due valori consecutivi differiscono al più di k. Ad esempio, $[1, 3, 2, 4]$ è 2-limitata, ma non è 1-limitata (perché $|1 - 3| > 1$, $|2 - 4| > 1$).

Data una sequenza A, una sottosequenza k-limitata di A è una sottosequenza di A che è anche k-limitata. Ricordiamo che le sottosequenze mantengono l'ordine della sequenza originale. Ad esempio, in $A = [\mathbf{4}, 6, \mathbf{3}, 1, \mathbf{4}]$ è presente la sottosequenza $[4, 3, 4]$, che è 1-limitata. Il valore di una sottosequenza è dato dalla somma dei suoi elementi. Nel caso di $[4, 3, 4]$, il suo valore è 11.

Scrivere un algoritmo

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

che prenda in input una sequenza A di valori positivi memorizzata in un vettore di dimensione n e un valore k, e restituisca il valore massimale fra tutte le sottosequenze k-limitata contenute in A. Ad esempio,

- Per $A = [4, 6, 3, 1, 4]$ e $k = 1$, la risposta è 11 ($[4, 3, 4]$)
- Per $A = [4, 6, 3, 1, 4]$ e $k = 2$, la risposta è 14 ($[4, 6, 4]$)
- Per $A = [4, 6, 3, 1, 4]$ e $k = 3$, la risposta è 18 ($[4, 6, 3, 1, 4]$)

Discutere correttezza e complessità computazionale dell'algoritmo proposto.