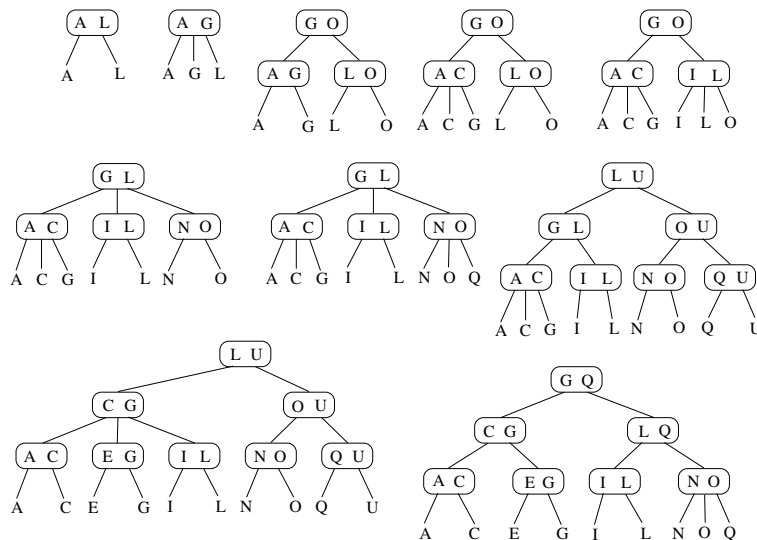


Nome: .....  
 Cognome: .....  
 Matricola: .....

1. Utilizzate i caratteri **ALGOCINQUE** per costruire un albero 2-3 attraverso inserimenti in sequenza (disegnate l'albero risultante al termine di ciascun inserimento). In ultimo, effettuate la cancellazione di **U** e mostrate l'albero finale.

*Sol.*



2. È possibile sviluppare una versione dell'algoritmo Quicksort che operi su lista (ovvero i dati da ordinare sono nei nodi di una lista concatenata) e che abbia le stesse prestazioni della versione basata su vettore? Motivate la risposta.

*Sol.* Il Quicksort è basato sull'operazione di partizionamento, la quale prevede la scelta di un elemento pivot  $p$  nella sequenza di ingresso, e la creazione di due sequenze di dati, la prima contenente i valori minori di  $p$ , la seconda contenente quelli maggiori. Se i dati sono contenuti in una lista, è sufficiente prendere il primo elemento come pivot e scandire tutti i successivi distribuendoli in due liste inizialmente vuote (quella per i minori  $L$  e quella per i maggiori  $G$ ) in funzione dell'esito del confronto con  $p$ . L'intera operazione (costruzione delle due liste  $L$  e  $G$ ) ha costo  $\Theta(n)$  (più precisamente, richiede  $n - 1$  confronti), ovvero lo stesso costo della versione su vettore. Di conseguenza, le equazioni di ricorrenza che descrivono il numero di confronti del Quicksort nei tre casi (migliore, peggiore e medio) sono le stesse della versione su vettore.

3. Date la definizione formale di albero binario di ricerca.

*Sol.* Un albero binario di ricerca per un insieme di dati  $V$  su cui è definita una relazione d'ordine totale  $<$ , è un albero in cui:

- ogni nodo contiene un elemento di  $V$ ;
- ogni nodo ha al massimo 2 figli (sinistro e destro);
- per ogni nodo dell'albero, se il nodo contiene un valore  $v$  allora il suo sottoalbero sinistro contiene unicamente valori minori di  $v$ , mentre il sottoalbero destro contiene unicamente valori maggiori di  $v$ .

4. Mostrate e commentate l'equazione di ricorrenza su cui è basata la soluzione di uno tra i seguenti problemi (a vostra scelta): chiusura transitiva di un grafo, cammini minimi in un grafo, moltiplicazione di matrici in cascata.

*Sol.* Si vedano i paragrafi 11.3, 11.4 e 11.5 delle dispense.

5. Scrivete un programma RAM che accetta in ingresso un intero  $n$  e restituisce  $n^n$ . Che complessità ha secondo il criterio di costo uniforme? E secondo il criterio di costo logaritmico?

*Sol.* Supponiamo che  $n$  sia maggiore di 0 e utilizziamo 3 registri oltre all'accumulatore.  $R_1$  per contenere il risultato,  $R_2$  per  $n$ ,  $R_3$  come contatore (inizializzato a  $n$ ). Si ha quindi

```
1 JBLANK 15
2 READ 0
3 STORE 1
4 STORE 2
5 STORE 3
6 LOAD 3
7 SUB =1
8 STORE 3
9 JZERO 14
10 LOAD 1
11 MULT 2
12 STORE 1
13 GOTO 6
14 WRITE 1
15 HALT
```

Secondo il criterio di costo uniforme, la complessità in spazio è pari al numero di registri utilizzati,  $S(n) = 4 = O(1)$ , mentre la complessità in tempo è  $T(n) = 11 + 8(n - 1) = \Theta(n)$ . Adottando il criterio di costo logaritmico, basta notare che il registro che contiene il valore più grande è  $R_1$  e che all'ultima iterazione contiene il valore  $n^n$  per cui vale  $\log n^n = n \log n$ . Dato che dopo  $\frac{n}{2}$  iterazioni  $R_1$  contiene  $n^{n/2}$  (con  $\log(n^{n/2}) = \frac{n}{2} \log n$ ), ciascuna delle ultime  $\frac{n}{2}$  iterazioni ha costo  $\Theta(n \log n)$ . La complessità secondo il criterio logaritmico è quindi  $S(n) = \Theta(n \log n)$  (spazio) e  $T(n) = \Theta(n^2 \log n)$  (tempo).