

### Spazio riservato alla correzione

1	2	3	4	Totale
/6	/8	/8	/10	/32

utilizzate altri fogli. Utilizzate soltanto lo spazio sottostante. Fogli differenti non saranno in considerazione per la correzione. Non scrivere a matita. Per tutti gli esercizi, descrivere la complessità asintotica delle funzioni implementate

Si consideri una coda **Q**, implementata con array **Q[MAX+2]**, **riempita con interi**. Si implementi la funzione ricorsiva **void moltiplica2(int Q[])** che, utilizzando una libreria di funzioni di accesso alla coda (**da implementare**) prendendo in input **Q**, restituisce la coda con gli elementi nello stesso ordine, sostituendo però ogni coppia di numeri consecutivi con il doppio del loro prodotto, dando priorità agli elementi più vicini al fine della coda. Si ricordi che la coda è una struttura dati che permette l'accesso ai suoi dati solo dalla testa.

Esempio: coda iniziale 3|5|7|4|5|5 (3 fine della coda) – coda finale 30|7|4|50|

Si considerino due liste di numeri interi **Lista1** e **Lista2** implementate come liste doppiamente puntate e non circolari, utilizzando la seguente struttura

```
struct elemento {  
    struct elemento *prev;  
    int inf;  
    struct elemento *next;}  
;
```

```
struct elemento *Lista1,*Lista2;
```

Si implementi una funzione **ricorsiva** che presi in input **Lista1** e **Lista2**

- rimuova da **Lista1** tutti i numeri pari
- rimuova da **Lista2** tutti i numeri dispari.
- rimuova da **Lista1** tutti i numeri multipli di 3 e li inserisca continuamente in testa alla **Lista2**.
- Restituisca le due liste modificate.

3. Siano T e P due alberi binari di ricerca, implementati con la seguente struttura a puntatori:

```
struct nodo {  
    int inforadice;  
    struct nodo *left;  
    struct nodo *right;}
```

```
struct nodo *T,*P;
```

Verificare che

- entrambi gli alberi sono degli ABR,
- entrambi hanno la stessa struttura
- dato n, la massima differenza tra le chiavi corrispondente tra i due alberi è al più n.



4. Siano **G** e **H** due grafi orientati pesati entrambi con pesi positivi, di **n** vertici  $0, 1, \dots, n-1$  e rappresentati con liste di adiacenza utilizzando la seguente struttura:

```
typedef struct graph {  
    int nv;  
    edge **adj; } graph;  
  
graph *G, *H;
```

```
typedef struct edge {  
    int key;  
    int peso;  
    struct edge *next; } edge;
```

- scrivere in linguaggio C una funzione che, presi in input **G** ne calcoli:
  - il grafo trasposto
  - il grado incidente e adiacente
- scrivere in linguaggio C una funzione che, presi in input **G** e **H** restituisca la differenza “esclusiva” di **G** e **H** nel seguente modo: per ogni arco in **G** presente anche in **H** si sottrae l'attuale peso in **G** con quello in **H** e se la differenza è negativa si elimina l'arco da **G**. Viceversa, se l'arco è in **H** ma non in **G**, si inserisce l'arco in **H** in **G**. Se invece è in **G** ma non in **H**, l'arco rimane invariato.

Dato un heap H (di interi) rappresentato con vettore statico, scrivere una procedura che, in tempo logaritmico, presi in input H e un valore numerico v presente in H (con relativo indice di posizione di v nel vettore), rimuova v da H, mantenendo la proprietà di heap.

Data una lista L dinamica singolarmente puntata di valori interi, raddoppiare (numericamente) tutti i numeri dispari in essa contenuti.

Per esempio 1->3->4 diventa 2->6->4

Data una lista L dinamica singolarmente puntata di valori interi, duplicare tutti i numeri dispari in essa contenuti (aggiungendo una copia a destra di ogni valore dispari).

Per esempio 1->2 diventa 1->1->2

Data una lista L dinamica singolarmente puntata di valori interi, per ogni coppia consecutiva di numeri dispari uguali, rimuovere un duplicato (il primo o il secondo della coppia).

Es. 1->1->1->2 diventa 1->1->2

Scrivere una funzione ricorsiva che permetta di creare due liste puntate e non circolari L1 e L2, di interi positivi, con caricamento da riga di comando

Scrivere una funzione ricorsiva per la stampa delle due liste

Scrivere una funzione ricorsiva che, prese entrambe le due liste L1 e L2

Ordini L1 in senso crescente

Rimuova da L1 gli elementi dispari e li inserisca in testa ad L2

Dal main, creare un menù che permetta di scegliere nell'ordine  
1 --> 2 --> 3 --> 2

Implementare tutto il codice necessario per l'algoritmo di Dijkstra per il calcolo del percorso minimo: creazione struttura dati grafo e strutture dati di appoggio - heap, interfaccia per il caricamento del grafo da tastiera, funzione stampa del grafo, implementazione dell'algoritmo e stampa del percorso minimo

Creare una funzione che permetta di restituire il percorso minimo tra il punto A e B passando necessariamente per un punto intermedio X. idea per la soluzione, prima si applica il Dijkstra da A a X e poi da X a B

Modificare la struttura dei grafi in modo che possa permettere di indicare se un arco  $(i, j)$  rappresenta una strada urbana o extraurbana. Di conseguenza, modificare il Dijkstra in modo che permetta di realizzare il percorso minimo tra due nodi nei seguenti 3 casi:

Generale, senza tener conto della tipologia delle strade

Utilizzando solo strade urbane

Utilizzando solo strade extraurbane



## Traccia LASD-Giugno 2022

Silvia Stranieri

Luglio 2022

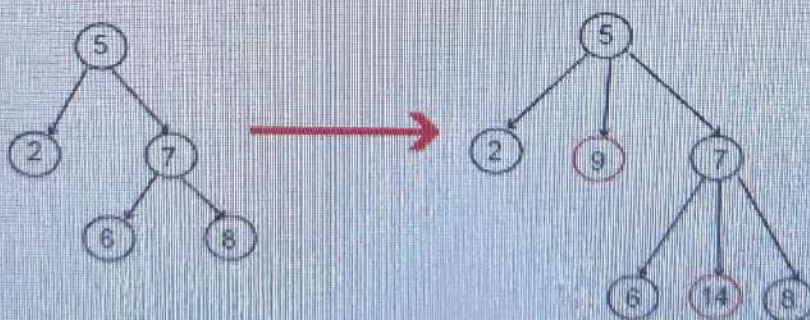
### 1 Esercizio 1: 10 punti

Scrivere una funzione che, data una coda  $Q$  di interi, modifichi  $Q$  in modo che sostituisca ad ogni coppia di valori pari consecutivi, il doppio della loro somma. La funzione restituisce la coda modificata.

Esempio:  $Q = 2|4|6|7|6|8, Q = 12|6|28$

### 2 Esercizio 2: 10 punti

Scrivere una funzione che, dato un ABR  $T$ , verifichi che  $T$  sia un ABR e costruisca un albero  $T'$  ternario in modo che, per ogni nodo in  $T$  che abbia entrambi i figli, si aggiunge in  $T'$  un terzo figlio *middle*, che è un nodo foglia, contenente la somma delle chiavi dei due fratelli.



Esempio:

### 3 Esercizio 3: 12 punti

Scrivere una funzione che, dati due grafi orientati pesati  $G$  e  $H$  e una lista  $L$ , crei un terzo grafo  $T$  risultante dall'unione di  $G$  e  $H$ , con la regola che l'arco



quale deve mostrare il funzionamento dell'esercizio.

#### Esercizio 1 [6]

Data una coda statica di elementi positivi, caricata da tastiera, scrivere una funzione che rimuove dalla coda tutte le occorrenze del numero 2. La funzione deve restituire la coda modificata e il suo minimo della coda.

#### Esercizio 2 [10]

Date due liste doppiamente puntate e non circolari di interi positivi L1 e L2, caricate da tastiera, scrivere una funzione in C che presi in input L1 e L2 rimuova da L1 tutte le occorrenze dei dispari e li inserisca in testa a L2.

Esempio: L1 = 1,3,4,5,6,3,4,7 ed L2=5 allora si ha L1 = 4,5,6, ed L2=7,3,5,3,1,5

#### Esercizio 3 [16]

Dati due grafi G1 e G2 entrambi orientati, pesati e di n vertici interi 0, 1, n-1, implementati con liste di adiacenza e caricati da tastiera. Scrivere una funzione in C che,

- Faccia la differenza dei due grafi in un terzo grafo G3 in modo che un arco (a,b) sarà presente con peso p in G3 se l'arco (a,b) è presente in G1 con peso p1, presente o meno in G2 con peso p2, e  $p = p1 - p2$  ed è positivo
- Scrivere l'algoritmo di Dijkstra per il percorso minimo sul grafo costruito.