Algoritmi e Strutture di Dati – A.A. 2022-2023
Esame scritto del 25/01/23 – 9CFU
Libri e appunti chiusi – Tempo = 2:00h

9

☐ Preferenze o indisponibilità per la data dell'orale (escluso dal 28/01 al 05/02 ed entro il 28/02).....
……
……
……

Cognome: _____ **Nome:** _____

DOMANDA SULLA COMPLESSITA' ASINTOTICA (3 punti su 30)

Discuti la complessità computazionale delle seguenti procedure nel caso peggiore fornendo O-grande, Omega e Theta in funzione del numero *n* di elementi dell'albero.

```
FUNZIONE(T)              /* T è un albero binario di interi */
      L.head = NULL      /* L è una nuova lista (vuota) di interi */
      FUNZ_RIC(T.root, L)
      return L

FUNZ_RIC(v, L)
      if( v==NULL ) return
      if( LUNGHEZZA(L) <= 10 )
            AGGIUNGI_IN_CODA(L,v.info)
      FUNZ_RIC(v.left, L)
      FUNZ_RIC(v.right, L)
```

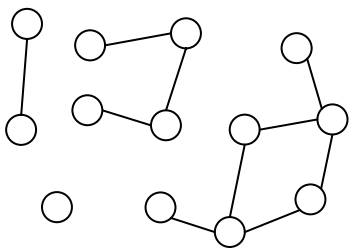
Assumi che sia **LUNGHEZZA** che **AGGIUNGI_IN_CODA** facciano un numero di operazioni proporzionale al numero degli elementi della lista.

ALGORITMO IN LINGUAGGIO C (27 punti su 30)

Scrivi in linguaggio C il codice della funzione

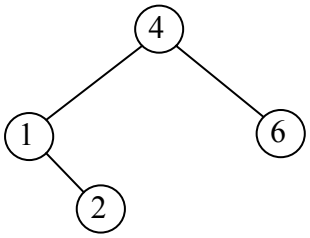
```
nodo_albero* abr_da_grafo(grafo* g)
```

che accetti in input un puntatore **g** ad un grafo non orientato rappresentato tramite oggetti e riferimenti in cui si assume (non deve essere verificato dalla funzione) che ogni componente connessa di **g** abbia un numero di nodi diverso dalle altre. La funzione restituisce in output un puntatore alla radice di un albero binario di ricerca (di altezza qualsiasi) che memorizza le dimensioni (il numero di nodi) delle componenti connesse di **g**. Se il grafo è NULL o non ha nessun nodo, la funzione ritorna NULL.



Grafo **g**

Possibile ABR in output



Per esempio nella figura a sinistra le componenti connesse del grafo **g** hanno dimensioni 1, 2, 4 e 6.

La funzione **abr_da_grafo**(g) potrebbe ritornare il puntatore all'albero binario di ricerca rappresentato a destra (ovviamente dipende dall'ordine con cui le componenti connesse vengono trovate).

Usa le seguenti strutture (che si suppone siano contenute nel file "strutture.h"):

```
typedef struct nodo_struct {
        elem_nodi* pos;  /* posizione nodo nella
                            lista del grafo */
        elem_archi* archi;   // lista archi incidenti
        int color;
} nodo;

typedef struct arco_struct {
        elem_archi* pos;  // pos. arco lista grafo
        nodo* from;
        nodo* to;
        elem_archi* frompos;  // pos. arco nodo from
        elem_archi* topos;      // pos. arco nodo to
} arco;

typedef struct elem_lista_nodi {
        struct elem_lista_nodi* prev;
        struct elem_lista_nodi* next;
        nodo* info;
} elem_nodi;   // elemento di una lista di nodi
```

```
typedef struct elem_lista_archi {
        struct elem_lista_archi* prev;
        struct elem_lista_archi* next;
        arco* info;
} elem_archi;  // elemento di una lista di archi

typedef struct {
        int numero_nodi;
        int numero_archi;
        elem_archi* archi;  // lista degli archi
        elem_nodi* nodi;    // lista dei nodi
} grafo;

/* struttura per l'albero binario di ricerca */

typedef struct nodo_albero_struct {
        struct nodo_albero_struct* left;
        struct nodo_albero_struct* right;
        int info;
} nodo_albero;
```

È possibile utilizzare qualsiasi libreria nota e implementare qualsiasi funzione di supporto a quella richiesta.