



**Laboratorio di Algoritmi e  
Strutture Dati**

Aniello Murano  
<http://people.na.infn.it/~murano/>

Murano Aniello - Lab. di ASD  
Nona e Decima Lezione - Mod.B

1



**Esercitazione di laboratorio:  
Problemi su grafi**

Murano Aniello - Lab. di ASD  
Nona e Decima Lezione - Mod.B

2

## Primo esercizio

- Siano  $G$  e  $H$  due grafi orientati pesati di  $n$  vertici  $0,1,\dots, n-1$  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;  
    struct edge *next; } edge;
```

scrivere in linguaggio C una funzione che restituisca un nuovo grafo  $T$  **unione** dei due grafi  $G$  e  $H$ , **rappresentato con liste di adiacenza**, secondo la struttura dati `graph` definita sopra. In pratica,  $T$  avrà tutti i vertici di  $G$  e  $H$  e conterrà un arco da un nodo  $i$  a un nodo  $j$  se tale arco è presente almeno in uno dei due grafi  $G$  e  $H$ . Per ogni arco aggiunto in  $T$ , se l'arco è presente solo in uno dei due grafi, allora l'arco eredita il peso dell'arco presente nel grafo di partenza. Se invece l'arco è presente in entrambi i grafi  $G$  e  $H$ , allora il suo peso sarà il minore tra i due pesi associati all'arco nei due grafi di partenza.



## Secondo esercizio

- Si consideri un grafo  $G$  orientato non pesato di  $n$  vertici  $0,1,\dots,n-1$ , rappresentato con liste di adiacenza secondo la struttura definita nel primo esercizio. Si scriva in linguaggio C una funzione che prenda in input il grafo  $G$  rappresentato con liste di adiacenza e restituisca un grafo  $T$  rappresentato con matrice di adiacenza. In pratica, la matrice  $T$  dovrà avere dimensione  $n \times n$  e dovrà essere riempita utilizzando la seguente regola:

per ogni vertice  $i, j \in n$ ,

$G[i][j]=1$  se esiste un arco da  $i$  a  $j$  in  $G$  e  $0$  altrimenti.

