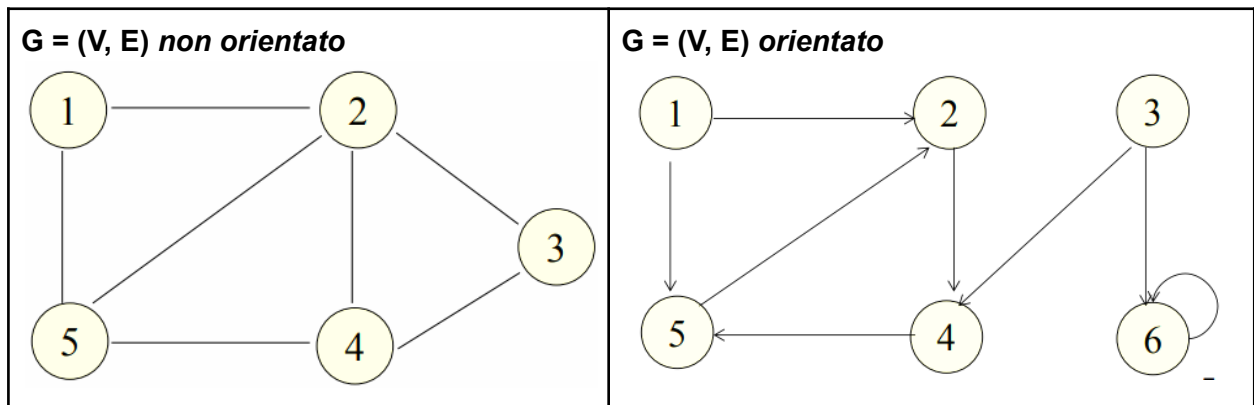


Grafì



Rappresentazione

Assumiamo che ogni vertice in V sia univocamente identificato da un identificativo nel range $[1, |V|]$ esistono 2 metodi standard per la rappresentazione

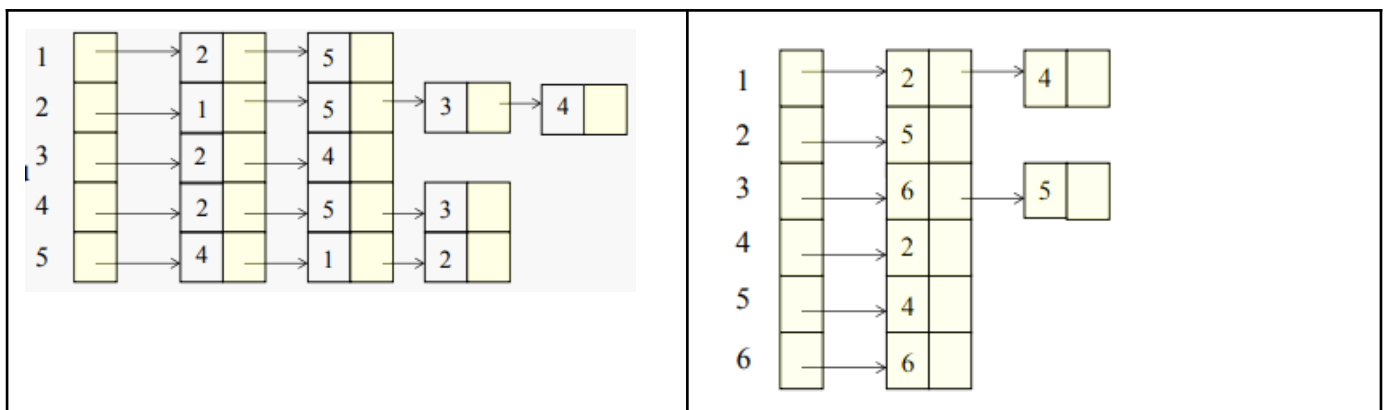
- Liste di adiacenza
- Matrici di adiacenza

Liste di adiacenza

- Array Adj di $|V|$ liste, una per ogni vertice in V .
- Per ogni vertice u in V , Adj[u] contiene tutti i vertici v in V tali che esista un arco (u,v) in E (tutti i vertici adiacenti a u in G , memorizzati in ordine arbitrario)
- Adj[u] contiene un puntatore alla testa della lista di tali vertici

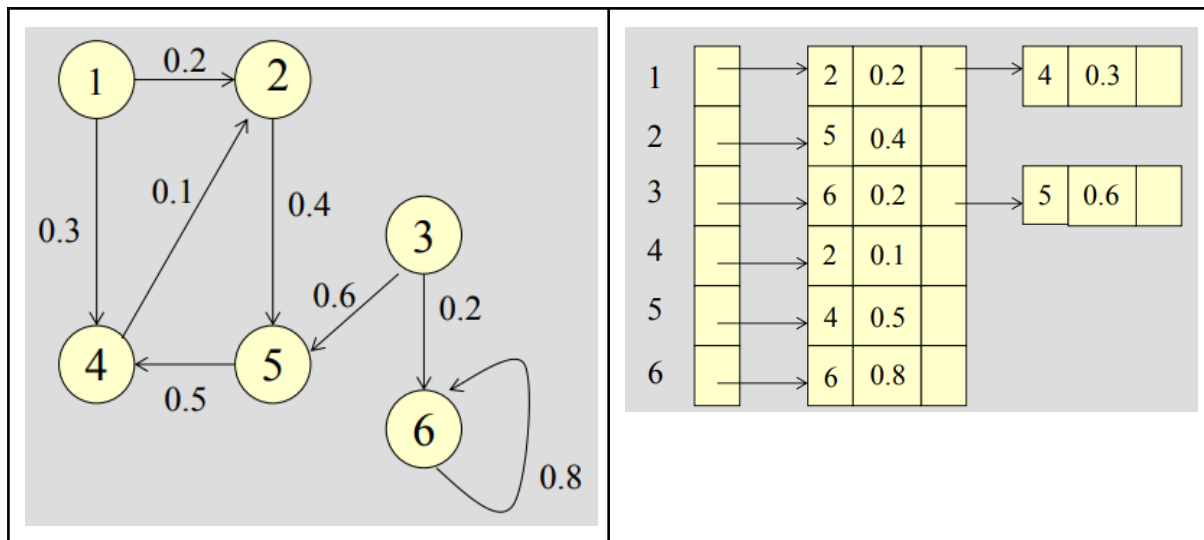
Ogni arco (u,v) è memorizzato nella lista di adiacenza di u e nella lista di adiacenza di v .

Esempio lista di adiacenza grafi riportati sopra



Peso in un grafo orientato

Il peso dell'arco (u,v) è memorizzato col vertice v nella lista u .

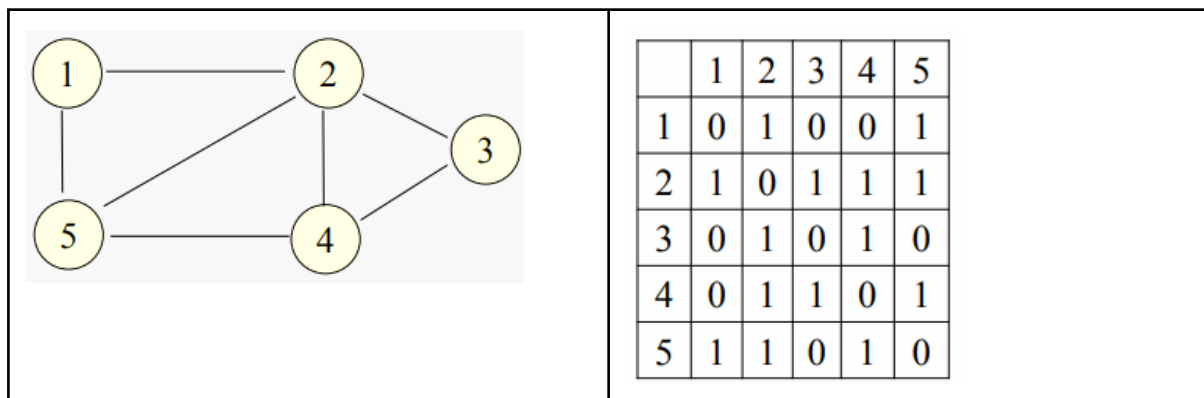


Matrici di adiacenza

Si utilizza una matrice $A = (a_{ik})$ di dimensione $|V| \times |V|$ dove $a_{ij} = \begin{cases} 1 & \text{se } (i,j) \text{ appartiene a } E \\ 0 & \text{altrimenti} \end{cases}$

Per archi pesati viene memorizzato il peso anziché il valore 1.

Esempio



Se mi riferisco all'i-esimo vertice devo fare $i + 1$.

Implementazione con liste

Struttura Lista di adiacenza

```
struct adj_node {
    int node;
    float weight;
    adj_node* next;
};
typedef adj_node* adj_list;
```

Puntatore al prossimo elemento della lista di adiacenza

Lista di adiacenza

Grafo

```
typedef struct {
    adj_list* nodes;
    int dim;
} graph;
```

Array dinamico di *dim* liste di adiacenze, una per vertice

Numero dei vertici

Le teste delle liste di adiacenza vengono memorizzate in un vettore dinamico della dimensione corrispondente al numero dei vertici.

```
g.nodes = new adl_list[g.dim]
```

L'identificativo dei nodi V di un grafo $G = (V, E)$ è nel range $[1, |V|]$.

Le primitive dovranno occuparsi della conversione da identificato del nodo a indice del vettore.

Se mi riferisco all' i -esimo vertice devo fare $i + 1$. ???

13->15

Lettura parametri passati all'eseguibile

int main(int argc, char *argv[])

- argc: numero di argomenti
- argv: contiene gli argomenti, uno per stringa salvati da argv[1] a argv[argc-1]. Per convenzione argv[0] contiene il nome con il quale il programma è stato invocato.

Esempio stampa tutti gli argomenti

```
int main(int argc, char *argv[])
{ /* main che stampa gli argomenti */
    for (int i = 0 ; i < argc ; i++)
        cout<<argv[i]<<endl ;
    return 0 ;
}
```