

TAD: GRAFO NON ORIENTATO CON VERTICI ETICHETTATI E ARCHI PESATI

STRUCT:

ABBIAMO 2 TIPI DI STRUCT:

- 1) "MEZZO ARCO", HA IL PUNTATORE DEL VERTICE DI DESTINAZIONE E IL PESO DELL'ARCO, TIENE INOLTRE IL PUNTATORE AL PROSSIMO MEZZO ARCO NELLA LISTA DI ADIACENZA
- 2) VERTICE, ABBIAMO: ETICHETTA DEL VERTICE, PUNTATORE ALLA LISTA DEI "MEZZI ARCHI" ADIACENTI, PUNTATORE AL PROSSIMO VERTICE E BOOLEANO PER VERIFICARE SE IL NODO E' STATO VISITATO.

1)

STRUCT HALFEDGEVERTEX {

LABEL	LABEL;	// ETICHETTA VERTICE DESTINAZIONE
VERTEX NODE*	VERPAT;	// PUNTATORE AL VERTICE
WEIGHT	WEIGHT;	// PESO DELL'ARCO
HALFEDGEVERTEX*	NEXT;	// MEZZO ARCO SUCCESSIVO

};

2)

STRUCT GRAPH: VERTEX NODE {

LABEL	LABEL;	// ETICHETTA VERTICE
HALFEDGEVERTEX*	ADJLIST;	// PUNTATORE LISTA MEZZI ARCHI ADIACENTI
VERTEX NODE*	NEXT;	// PROSSIMO VERTICE
BOOL	IS VISITED;	// BOOLEANO PER VERIFICARE SE IL NODO E' VISITATO

};

ADD VERTEX

IL PRIMO CONTROLLO DA FARE E' QUELLO DI VERIFICARE SE IL GRAFO E' VUOTO. NEL CASO DI GRAFO VUOTO AGGIUNGERE UN NUOVO VERTICE E' RADICE:

CREIAMO UN NUOVO NODO TRAMITE: `VERTEX NODE *AUX = NEW VERTEX NODE;`

CI BASTA POI ASSEGNARE TUTTI I CAMPI E VERIFICARE ANCHE `G = AUX`

INSERIRE, ASSIEME ALLE ALTERE, IL PUNTO GIÀ, INSERIRE IN CODA.

ADD EDGES

LA FUNZIONE SI OCCUPA DI INSERIRE UN ARCO TRA DUE VERTICI.
CONTINUANO INARZUITO SE I DUE NODI ESISTONO, PER CASCADUE PUNTATORI
E GLI ASSEGNA A NULLPTR.

PER CONTINUARE, COME TUTTO G E SE $CUR \rightarrow LABEL == FROM$ ALORA
NODO 1 = CUR, STESSA COSA CON NODO 2.

SE UNO DEI DUE (O ENTRAMBI) SONO ALORA NULLPTR, RETUR FALSE, SENNO
DEVO CONTINUARE SE ESISTE GIÀ UN ARCO:

```
HALFEDGEVERTEX * ARCO = NODO1 -> ADJLIST;
```

```
WHILE (ARCO != NULLPTR) {
```

```
    IF (ARCO -> LABEL == NODO2 -> LABEL) {
```

```
        RETUR FALSE;
```

```
    }
```

```
    ARCO = ARCO -> NEXT;
```

```
}
```

SE NON ESISTE ALORA POSSO PROCEDERE CON LA CREAZIONE:

```
HALFEDGEVERTEX * ARCO1 = NEW HALFEDGEVERTEX;
```

```
ARCO1 -> LABEL = NODO2 -> LABEL
```

```
ARCO1 -> VERTICI = NODO2;
```

```
ARCO1 -> WEIGHT = W;
```

```
ARCO1 -> NEXT = NODO1 -> ADJLIST;
```

```
NODO1 -> ADJLIST = ARCO1;
```

↑

STESSO PROCEDIMENTO CON ARCO2.

NUM EDGES

QUESTA FUNZIONE HA IL COMPITO DI RITORNARE IL NUMERO DI
ARCHI DEL GRAFO.

PER FARLO ABBIAMO BISOGNO DI DUE CICLI, UNO PER CUN != NULL

E L'ALTRO PER L'ARCO, ASSEGNIAMO HALFEDGEVERTEX * ARCO = CUN -> ADJLIST

WHILE (G != NULL) {

VERTEXNODE * TEMP = G;

G = G → NEXT;

HALFEDGE VERTEX * ADJ = TEMP → ADJLIST;

WHILE (ADJ != NULL) {

HALFEDGE VERTEX * TEMPADJ = ADJ;

ADJ = ADJ → NEXT;

DELETE TEMPADJ;

}

DELETE TEMP;

}

G = EMPTYGRAPH;

LIBERA LA LISTA DI ADIACENTI
DEL VERTECE CORRENTE.

PRINT

VERTEXNODE * CUR = G;

WHILE (CUR != NULL) {

COUT << CUR → LABEL << " ";

HALFEDGE VERTEX * ADJ = CUR → ADJLIST;

BOOL FIRST = TRUE

SENZA PER VERIFICARE SE SI INIZIA DA UN

WHILE (ADJ != NULL) {

IF (!FIRST) {

COUT << " , ";

}

COUT << " (" << ADJ → LABEL << " (" << ADJ → WEIGHT <<

FIRST = FALSE;

ADJ = ADJ → NEXT;

}

COUT << ENDL;

CUR = CUR → NEXT;