Francesco Rollo s273802
L09E01 “Grafi e DAG”

Le **strutture dati** utilizzate sono:

- **Quasi ADT** “Item” contenente una stringa sovradimensionata ***node*** e “Key” definiti in “**Item.h**”.
  Sul tipo Item sono definite le funzioni di caricamento (ITEMload), stampa (ITEMstore), generazione di un dato vuoto (ITEMsetvoid).
  Sul tipo Key sono definite le funzioni di confronto (KEYcmp) e di estrazione della chiave (KEYget);

- **Quasi ADT** “Edge” contenente 3 interi **v**, **w** e **wt** che indicano i vertici su cui insiste ogni arco e il suo relativo peso. Edge è definito in “**Graph.h**”.
  Sul tipo Edge è definita la funzione di stampa (EDGEstore);

- **ADT 1° CLASSE** “ST” nonché una tabella di simboli con vettore ordinato definita in “**ST.h**”. Il file “ST.c” contiene la definizione della struct per il wrapper che contiene il vettore di tipo **Item** e 2 interi **M** ed **N** che descrivono il numero massimo e corrente di elementi.
  Sul tipo ST sono definite le funzioni di inizializzazione (STinit), inserimento (STinsert), ricerca (STsearch), stampa (STdisplay) e distruzione (STfree);

- **ADT 1° CLASSE** “Graph” che rappresenta un grafo orientato e pesato tramite la matrice delle adiacenze definito in “**Graph.h**”. Nel file “Graph.c” viene definita la struct contenente i due interi **V** ed **E**, la matrice **madj** e la tabella di simboli **st**.
  Sul tipo Graph sono definite le funzioni di inizializzazione (GRAPHinit), caricamento (GRAPHload), stampa (GRAPHstore), distruzione (GRAPHfree), inserimento e rimozione di un arco (GRAPHinsertE e GRAPHremoveE), caricamento degli archi (GRAPHedges).
  Sono definite poi le funzioni relative ai DAG nonchè rilevazione di un DAG tramite ricerca in profondità (TSdfsR), ordinamento topologico (DAGrts), trasformazione in DAG (createDAG) e calcolo delle distanze massime di ogni nodo (GRAPHmaxDist).

Per capire se il grafo del file è ciclico o aciclico ho effettuato una ricerca in profondità tramite la funzione **TSdfsR** chiamata dalla funzione **DAGrts** nella quale mi passo un booleano type aggiornato in base alla presenza di un DAG o meno. In realtà questa funzione mi permette anche di generare il vettore ***ts*** dove per ciascun tempo si registra quale vertice è stato completato e poter effettuare in seguito l’ordinamento topologico del grafo.

Per quanto riguarda i primi due punti, nella creazione del DAG ho definito due elementi importanti ovvero il numero massimo di archi removibili dato da E-(V-1) e un modello combinatorio basato sulle combinazioni semplici per individuare la ***bestsol*** (vettore di indici contente la posizione dei vertici) a peso maggiore da applicare nella rimozione degli archi.

Nell’ultimo punto per calcolare le distanze massime da ogni nodo, tramite due cicli concatenati calcolo lungo tutti i vertici per ogni vertice la distanza dei restanti vertici a esso collegati tramite l’utilizzo del vettore st e la matrice delle adiacenze per verificare la connessione.