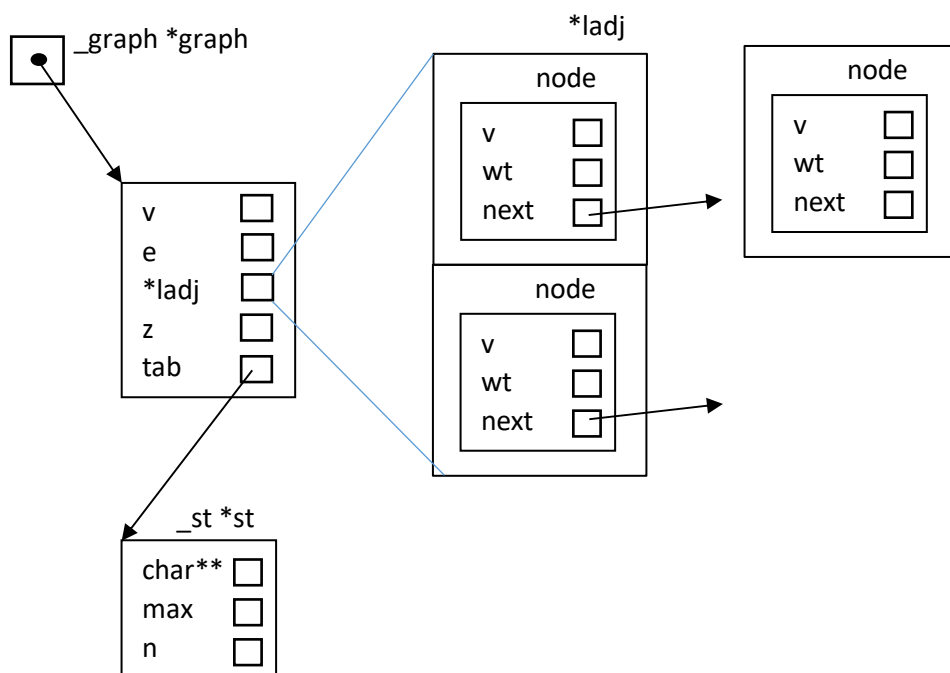


Michele Cerra, S259105

Relazione su: LAB 12

- Per la risoluzione del problema ho usato un grafo con lista di adiacenza, e con tabella di simboli creata con vettore di stringhe.
Ho scelto di fare la tabella di simboli con un vettore perché nel file di input viene fornito il numero di vertici, perciò possiamo allocare un vettore non sovradimensionato.
Per quanto riguarda le stringhe di massimo 30 caratteri, ho deciso di fare anche loro dinamiche, perciò il controllo della dimensione viene fatto in GRAPHload, dove è presente un vettore di caratteri statico.



- Prima di cominciare controllo se il grafo appena caricato sia già un DAG, se questo è vero allora vado direttamente al 3 punto del LAB, altrimenti comincio a fare la ricerca degli archi di cardinalità minima.
Per fare ciò ho usato il modello powerset con combinazioni semplici, partendo da cardinalità 1 a salire. Attraverso la funzione GRAPHedges creo un vettore di archi del grafo, e uso questo vettore come valori principali per il powerset. Per ogni insieme di archi trovato, vedo se rimossi dal grafo formano un DAG, alla prima ricerca positiva non con continuo con powerset di cardinalità superiore, faccio un confronto del peso massimo, se supera il massimo allora me li salvo in una `b_sol`, che mi servirà poi dopo per rimuoverli definitivamente dal grafo, intanto continua la ricerca del insieme e la stampa.
Una volta trovato il max e averlo rimosso dal grafo, il grafo diventa un DAG per definizione.
Per il calcolo delle distanze massime, faccio prima una ricerca dei vertici sorgente e dell'ordine topologico inverso del grafo, adesso per ogni sorgente cerco le massime distanze da ogni altro vertice raggiungibile.
L'algoritmo usato parte dal vertice sorgente nell'ordine topologico inverso e scandendolo al contrario verifico che per ogni vertice sia verificata la condizione:
$$d[v] < d[u] + w(u, v)$$

Se è vero allora aggiorno la distanza massima. Infine stampo tutto.