

All together

Scrivere in un file `Graph.cpp` le implementazioni delle funzioni definite all'interno del file `Graph.h`. I file `Graph.h` e il `main` del programma possono essere scaricati dalla cartella `graph` su Google Drive. Lo scopo è quello di trovare, dato un grafo di dipendenze, tutte le dipendenze di un nodo. Per farlo sarà necessario implementare un `set`.

Definizioni:

- **Dependency Graph**

Un grafo orientato, dove un arco $A \rightarrow B$ rappresenta la dipendenza di A da B. Nel caso B avesse altre dipendenze, allora A dipenderà anche da loro, ecc. . .

- **Set**

Il `set` è una struttura dati che, in questo caso, è un array dinamico la cui dimensione varia in base al numero di elementi contenuti. Gli elementi al suo interno devono rimanere sempre in ordine crescente e non sono ammessi duplicati.

Quando il `set` è pieno e si cerca di inserire un nuovo elemento, la lunghezza dell'array deve essere moltiplicata per `expansion_factor = 2`.

Quando la percentuale di carico dell'array diventa inferiore a `LOAD = 0.25`, allora la dimensione deve essere divisa per `expansion_factor` (deallocando correttamente la memoria non più in uso).

NOTE:

- il `main` del programma può essere modificato per testare il `set` prima di utilizzarlo nella ricerca delle dipendenze di un nodo all'interno del grafo
- il file `Graph.h` non può essere modificato
- per comodità, è ammesso l'uso della struttura dati `stack` fornita dalla libreria di sistema (per avere uno stack di variabili `int` usare `stack<int> nome;`, servirà per la visita del grafo).
- i nodi e gli archi del grafo devono essere letti da un file di testo come quello fornito (`graph1.txt`) dove la prima riga contiene il numero di nodi del grafo e le righe successive gli archi.

Ad esempio `0 1` indica un arco `0 -> 1`

Estraendo le dipendenze in `graph1.txt` si ottiene una rappresentazione come lista di adiacenza

```
0 -> [1, 2, 3, 4, 5]
1 -> [2, 4, 5]
2 -> []
3 -> [0, 1, 2, 4, 5]
4 -> [2]
5 -> [2, 4]
```

oppure come matrice di adiacenza

	0	1	2	3	4	5
0	-	1	1	1	1	1
1	0	-	1	0	1	1
2	0	0	-	0	0	0
3	1	1	1	-	1	1
4	0	0	1	0	-	0
5	0	0	1	0	1	-

si possono rappresentare le precedenti strutture graficamente come

