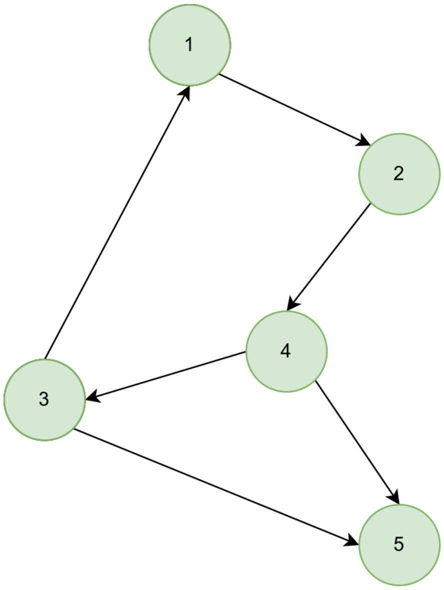
Lab 5 - Graph Representations, Depth First Search, Breadth First Search



1. Implement the following graph operations using an adjacency matrix:
   (a) Initialise a graph with n nodes.★
   (b) Insert an edge in the graph ★
   (c) Delete an edge from the graph ★
   (d) Print the graph ★★
   (e) Deallocate the graph (free the memory for the nodes and the adjacency matrix) ★
   (f) Depth first search ★★
   (g) Breadth first search ★★

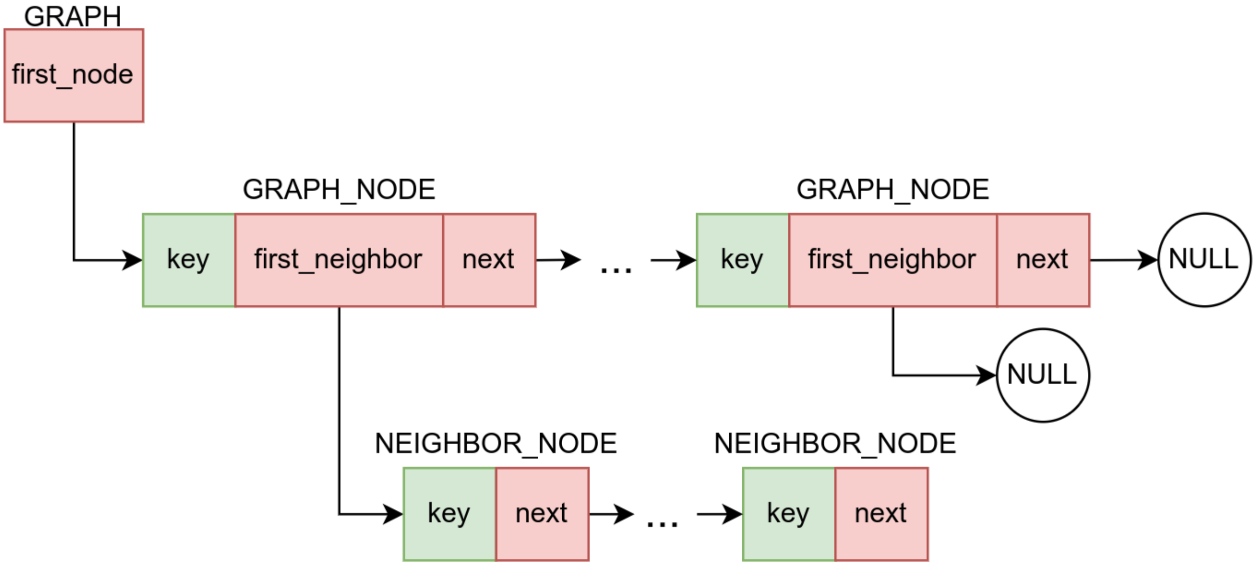
2. Implement the following graph operations using an edge list:
   (a) Initialise a graph with n nodes.★
   (b) Insert an edge in the graph ★★
   (c) Delete an edge from the graph ★★
   (d) Print the graph ★★
   (e) Deallocate the graph (free the memory for the nodes and the edge list) ★★
   (f) Depth first search ★★
   (g) Breadth first search ★★

3. Implement the following graph operations using a data strucure similar to the following example:

```
typedef struct _NEIGHBOR_NODE{
    int key;
    struct _NEIGHBOR_NODE* next;
}NEIGHBOR_NODE;

typedef struct _GRAPH_NODE{
    int key;
    NEIGHBOR_NODE* first_neighbor;
    struct _GRAPH_NODE* next;
} GRAPH_NODE;

typedef struct{
    GRAPH_NODE* first_node;
}GRAPH;
```



Create a function for each of the following operations:

(a) Initialise the graph ★

(b) Insert a node in the graph ★★

(c) Delete a node from the graph ★★

(d) Print the graph ★★

(e) Insert an edge in the graph ★★

(f) Delete an edge from the graph ★★

(g) Deallocate the graph (free the memory for the nodes and the graph) ★★

(h) Depth first search ★★★

(i) Breadth first search ★★★

Note: Leave a comment with the text PB1, PB2.A.II, ... PB10 above every function that implements the respective lab task. (upper case text, no space between the text and the problem number)