

جامعة المنارة

كلية: الهندسة

قسم: الهندسة المعلوماتية

اسم المقرر: الخوارزميات و بني المعطيات 2

رقم الجلسة (الأولى)

عنوان الجلسة

تمثيل البيان

Graph Representations



العام الدراسي 2023 - 2024

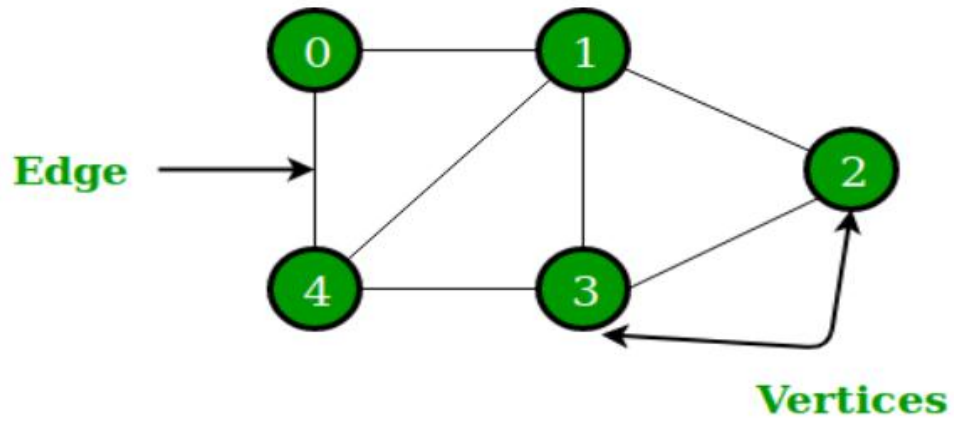
الفصل الدراسي الثاني

تمثيل البيان

الغاية من الجلسة:

- ✓ تمثيل البيان باستخدام مصفوفة التجاور وقوائم التجاور .
- ✓ تنفيذ الكود الخاص بتمثيل البيان بقوائم التجاور .

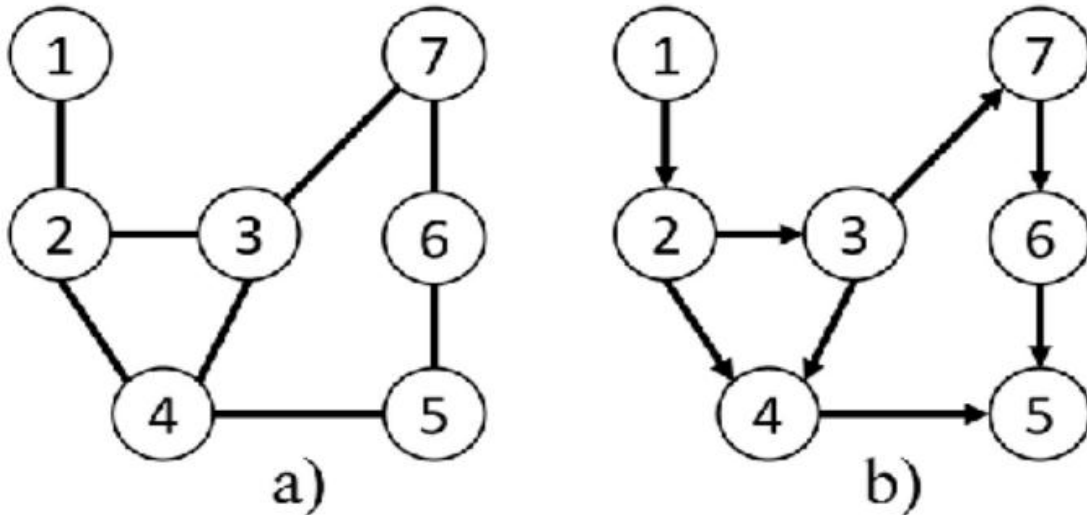
تعريف البيان :



يتم تعريف البيان G بالثنائية $G=(V,E)$ ، حيث تمثل V مجموعة رؤوس البيان ، و E مجموعة حواف البيان .

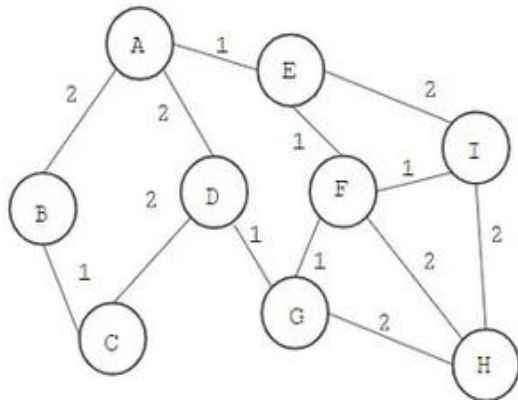
: Directed graph and Undirected Graph البيان الموجه و الغير موجه

يمثل البيان في الشكل (a) بيان غير موجه ، أما البيان في الشكل (b) فهو بيان موجه .

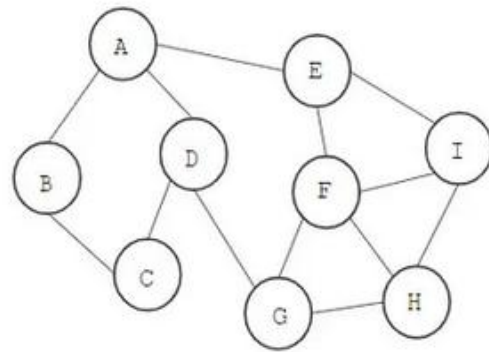


البيان الموزون و الغير موزون Weighted graph and Unweighted Graph :

Weighted graph



Unweighted graph

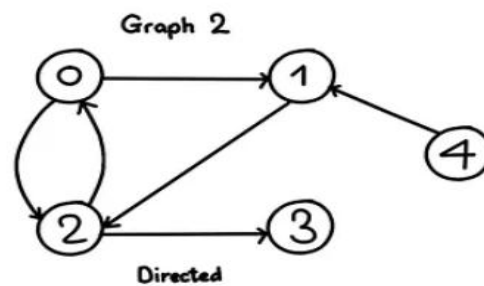
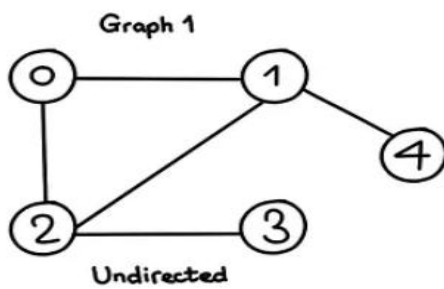


تمثيل البيان:

يتم تمثيل البيان بطريقتين:

1. التمثيل بمصفوفة التجاور Adjacency Matrix

تمرين 1 : قم بتمثيل البيان بمصفوفة التجاور .



Graph 1

adjacency matrix

	0	1	2	3	4
0	0	1	1	0	0
1	1	0	1	0	1
2	1	1	0	1	0
3	0	0	1	0	0
4	0	1	0	0	0

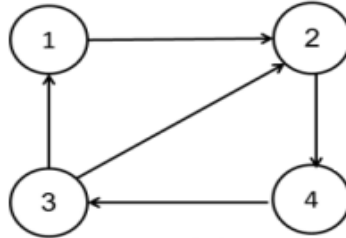
Graph 2

adjacency matrix

	0	1	2	3	4
0	0	1	1	0	0
1	0	0	1	0	0
2	1	0	0	1	0
3	0	0	0	0	0
4	0	1	0	0	0

2. التمثيل بقوائم التجاور Adjacency Lists

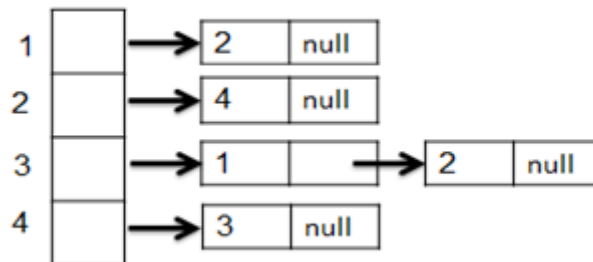
تمرين 2: ليكن لدينا البيان التالي والمطلوب : قم بتمثيل البيان باستخدام مصفوفة التجاور وقوائم التجاور.



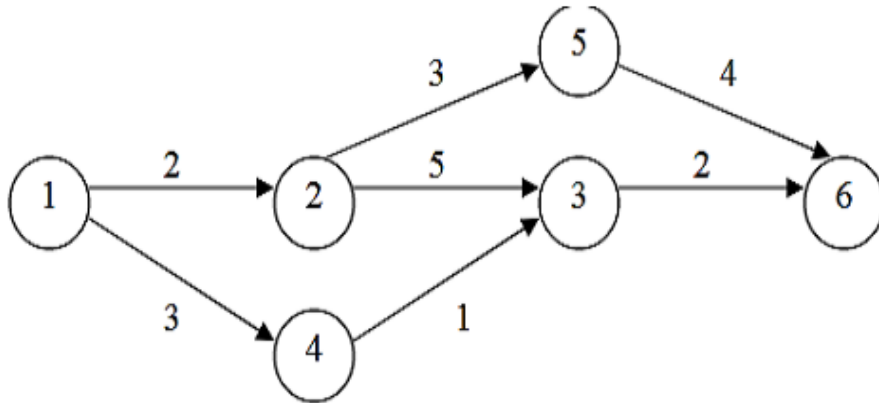
مصفوفة التجاور للبيان :

	1	2	3	4
1	0	1	0	0
2	0	0	0	1
3	1	1	0	0
4	0	0	1	0

قوائم التجاور للبيان :



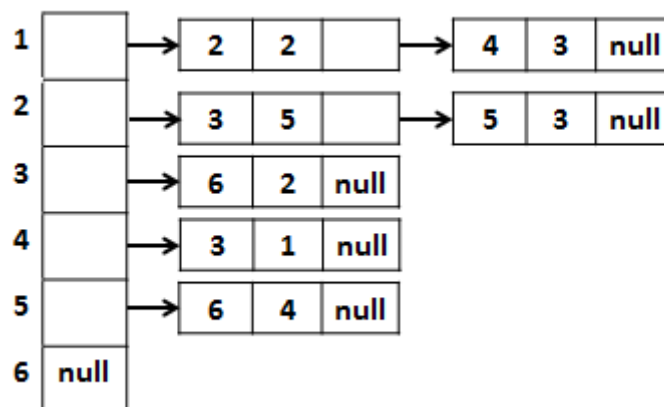
تمرين 3 : قم بتمثيل البيان التالي باستخدام مصفوفة التجاور وقوائم التجاور.



مصفوفة التجاور للبيان :

	1	2	3	4	5	6
1	0	2	∞	3	∞	∞
2	∞	0	5	∞	3	∞
3	∞	∞	0	∞	∞	2
4	∞	∞	1	0	∞	∞
5	∞	∞	∞	∞	0	4
6	∞	∞	∞	∞	∞	0

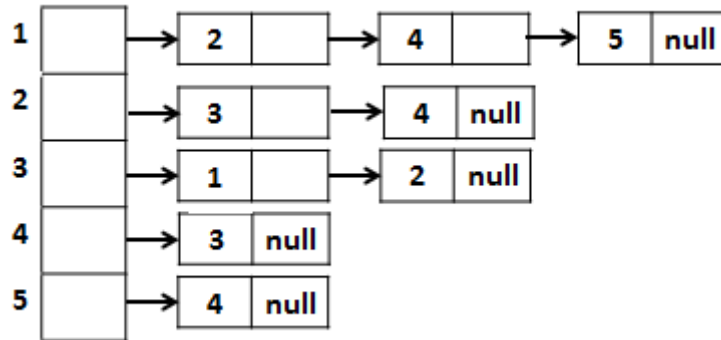
قوائم التجاور للبيان :



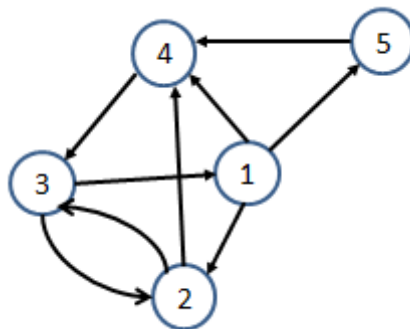
تمرين 4 : ليكن لدينا بيان يحوي 5 عقد ممثل بمصفوفة التجاور التالية، والمطلوب مثل البيان بقوائم التجاور وارسم البيان .

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

قوائم التجاور للبيان :



البيان :



البرنامج الخاص بتمثيل البيان باستخدام قوائم التجاور :

إن بنية المعطيات التي تستخدم لتمثيل قوائم التجاور برمجياً هي القوائم الوصلية linked list .

```
#include <iostream>
using namespace std;
// stores adjacency list items
struct adjNode {
    int val;
    adjNode* next;
};
// structure to store edges
struct graphEdge {
    int start_ver, end_ver;
};
class DiaGraph{
    // insert new nodes into adjacency list from given
graph
    adjNode* getAdjListNode(int value, adjNode* head)    {
        adjNode* newNode = new adjNode;
        newNode->val = value;

        // point new node to current head
        newNode->next = head;

        return newNode;
    }
    // number of nodes in the graph
public:
    adjNode **head;
    int N;
    //adjacency list as array of pointers
    // Constructor
    DiaGraph(graphEdge edges[], int n, int N)    {
        // allocate new node
        head = new adjNode*[N+1];
        this->N = N;
        // initialize head pointer for all vertices
        for (int i = 1; i <= N; ++i)
            head[i] = NULL;
        // construct directed graph by adding edges to it
        for (int i =0; i < n; i++)    {
            int start ver = edges[i].start ver;
```

```

        int end_ver = edges[i].end_ver;

        // insert in the beginning
        adjNode* newNode = getAdjListNode(end_ver,
head[start_ver]);

        // point head pointer to new node
        head[start_ver] = newNode;
    }
}

};
// print all adjacent vertices of given vertex
void display_AdjList(adjNode* ptr, int i)
{
    cout <<"adjacency list of vertex(" <<i<<"): ";
    while (ptr != NULL) {
        cout << "(" << i << ", " << ptr->val << " ) ";
        ptr = ptr->next;
    }
    cout << endl;
}
// graph implementation
int main()
{
    // graph edges array.
    graphEdge edges[] = {{1,2},{1,3},{2,1},{3,2}};
        // (x, y) -> edge from x to y

    int N =3;        // Number of vertices in the graph
    // calculate number of edges
    int n = sizeof(edges)/sizeof(edges[0]);
    // construct graph
    DiaGraph diagraph(edges, n, N);
    // print adjacency list representation of graph
    cout<<"Graph adjacency list "<<"(start_vertex,
end_ver):"<<endl<<endl;
    for (int i = 1; i < N+1; i++)
    {
        // display adjacent vertices of vertex i
        display_AdjList(diagraph.head[i], i);
    }
    return 0;
}

```


تمرين غير محلول:

: ليكن لدينا بيان يحوي 7 عقد ممثل بمصفوفة التجاور التالية :

	1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	0	0	0	1	0	0	0
3	0	0	0	0	1	1	0
4	1	0	0	0	1	0	0
5	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1
7	0	0	0	0	1	0	0

والمطلوب :

A. مثل البيان بقوائم التجاور .

B. ارسم البيان .