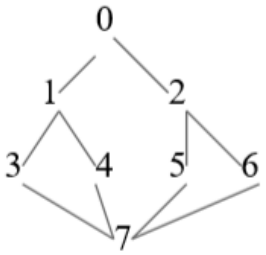
# Lab11 Graph Search - (BFS)

- BFS 테스트는 배열 기반

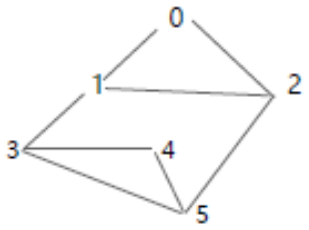
  int graph[max][max]={{0,0,0,0,1,1}{0,0,1,1,0,1}.....}

  int front=0,rear=0, queue[5]; char visited[max];

- Data for BFS (다음 2 개의 그래프로 테스트 할 것)




(Graph 1)




(Graph 2)

- 위 그래프 입력 데이터는 Adjacency Matrix 로 구성할 것

******* Adjacent Matrix**

| | v0 | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v0 | v1 | v2 | v3 | v4 | v5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **v0** | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| **v1** | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| **v2** | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| **v3** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| **v4** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| **v5** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| **v6** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | | | | | | |
| **v7** | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | | | | | | |

- **Output ( BFS 탐색의 결과)**

- **첫번째 데이터 : 0 1 2 3 4 5 6 7**
- **두번째 데이터 : 0 1 2 3 5 4**

- **알고리즘**

```
void main()
{    initializeQ( );                  // Queue 초기화   front=rear = 0;
     print Adjacent Matrix;     // 그래프 데이터 출력
     bfs(v);   // 첫번 노드부터 시작.
}

bfs() {
. Initialize visited[];        // visited 'false'로 초기화
. addq(v);                     // v-> 시작 정점
. v= deque()
. while (!Que-empty) {
   for (인접된 모든 노드 w 에 대해서)
       if (not visited)&& (graph[v][w] !=0) {     // 방문되지 않았고 & 0이 아니면
              addq(w);                  //
               visited[w] = 'true';              // marking
               cout << "    " << w;          // 출력
        }
    v = deletequeue();}      // get next node;
}
```

- Other ADT
  - AddQ,  Delete Q,  Print Matrix

- 실행 화면