

ITM 517 Algorithm

Ja-Hee Kim

Graph





Traversal

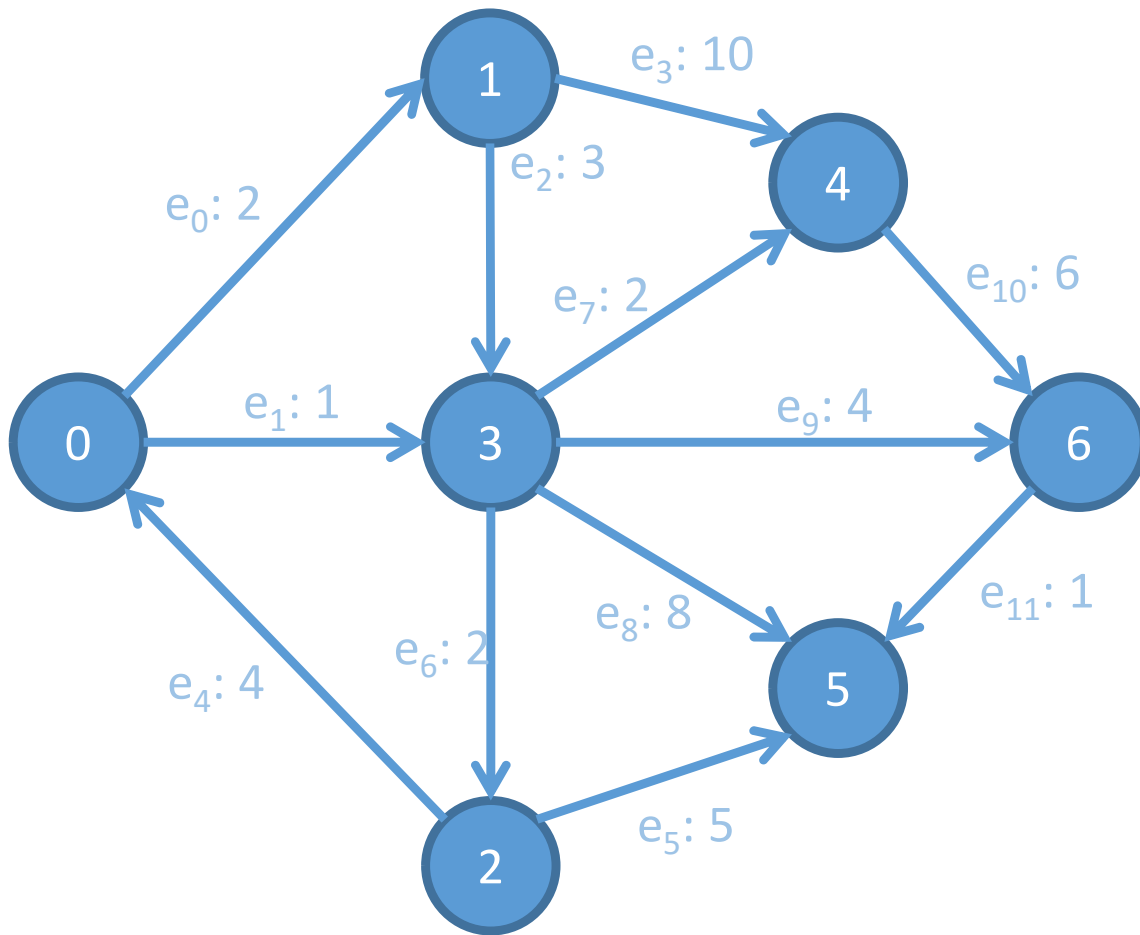
Graph traversal

- A graph traversal begins at a any vertex (origin vertex) and visits only the vertices that it can reach.
 - Depth First Traversal
 - Breadth First Traversal

Depth First Traversal

- A depth-first traversal visits a vertex, then a neighbor of the vertex, a neighbor of the neighbor, and so on, advancing as far as possible from the original vertex.
- Algorithm for given origin
 1. Define traversalOrder
 2. Put origin to traversalOrder and marked as visited
 3. Check all the neighbors of origin
Dfs(the neighbor)

Example of DFT



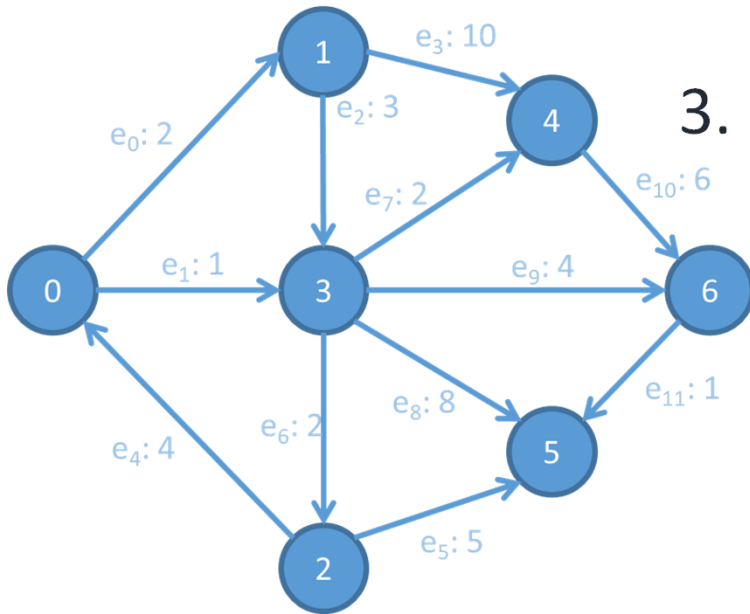
1. Define traversalOrder
2. Put origin to traversalOrder and marked as visited
3. Check all the neighbors of origin
Dfs(the neighbor)

TraversalOrder:

Breadth First Traversal

- A breadth first traversal visits a vertex and then each of the vertex's neighbors before advancing.
- Algorithm
 1. Make queues for traversalOrder, vertexQ
 2. Enqueue the origin vertex to traversalOrder and vertexQ
 3. While (!vertexQ.isEmpty)
 1. frontVertex = vertexQ.dequeue
 2. While(frontVertex has a neighbor)
 1. nextNeighbor = nextNeighbor of frontVertex
 2. Mark nextNeighbor as visited
 3. If(nextNeighbor is not visited)
 1. Mark nextNeighbor as visited
 2. TraversalOrder.enqueue(nextVertex)
 3. vertexQ.enqueue(nextVertex)

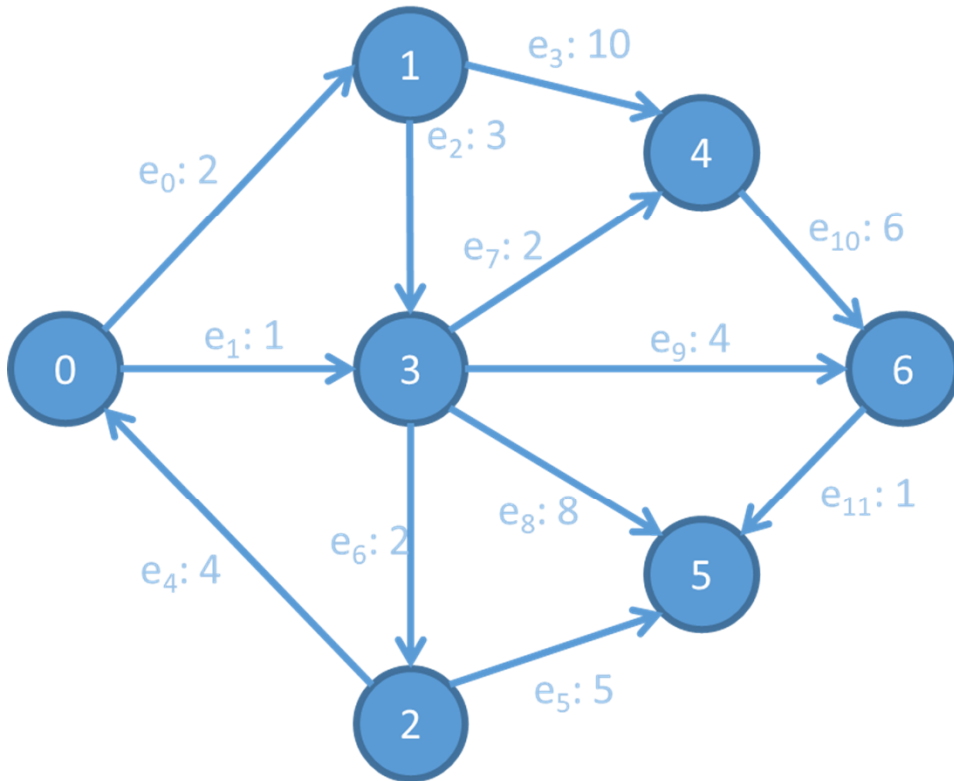
Example of BFT



1. Make queues for traversalOrder, vertexQ
2. Enqueue the origin vertex to traversalOrder and vertexQ
3. While (!vertexQ.isEmpty)
 1. frontVertex = vertexQ.dequeue
 2. While(frontVertex has a neighbor)
 1. nextNeighbor = nextNeighbor of frontVertex
 2. Mark nextNeighbor as visited
 3. If(nextNeighbor is not visited)
 1. Mark nextNeighbor as visited
 2. TraversalOrder.enqueue(nextVertex)

traversalOrder	
vertexQ	

Implementation of traversal



```
1 public class GraphTest {
2     public static void main(String[] args) {
3         Graph g = new AdjMatrix(7);
4         g.addEdge(0, 3, 1);    g.addEdge(0, 1, 2);
5         g.addEdge(1, 4, 10);   g.addEdge(1, 3, 3);
6         g.addEdge(2, 5, 5);    g.addEdge(2, 0, 4);
7         g.addEdge(3, 6, 4);    g.addEdge(3, 5, 8);
8         g.addEdge(3, 4, 2);    g.addEdge(3, 2, 2);
9         g.addEdge(4, 6, 6);    g.addEdge(6, 5, 1);
10
11         System.out.println(g);
12
13         System.out.println("BFT: " + g.bft(0));
14         System.out.println("DFT: " + g.dft(0));
15     }
16 }
```

Console Problems Debug Shell

<terminated> GraphTest (1) [Java Application] C:\WProgram Files\Java\jdk-15.0.1\bin\javaw.exe (2021. 3. 22. 오후 11:04:31 - 오후 1)

0.0	2.0	Infinity	1.0	Infinity	Infinity	Infinity
Infinity	0.0	Infinity	3.0	10.0	Infinity	Infinity
4.0	Infinity	0.0	Infinity	Infinity	5.0	Infinity
Infinity	Infinity	2.0	0.0	2.0	8.0	4.0
Infinity	Infinity	Infinity	Infinity	Infinity	0.0	Infinity
Infinity	Infinity	Infinity	Infinity	Infinity	Infinity	0.0
Infinity	Infinity	Infinity	Infinity	Infinity	Infinity	1.0
						0.0

BFT: 0 1 3 4 2 5 6
DFT: 0 1 3 2 5 4 6

AdjMatrix BFT

```
public String bft(int n) {  
    String s = "";  
    boolean[] visit = new boolean[m.length];  
    for(int i = 0; i < m.length ; i++)  
        visit[i] = false;  
    Queue<Integer> q = new ArrayDeque<Integer>();  
    q.add(n);  
    visit[n] = true;  
    Integer temp;  
    while(q.size() > 0){  
        temp = q.remove();  
        s += temp + " ";  
        for(int i = 0 ; i < m.length ; i++) {  
            if(m[temp][i] < Double.POSITIVE_INFINITY)  
                if(visit[i] == false) {  
                    q.add(i);  
                    visit[i] = true;  
                }  
        }  
    }  
    return s;  
}
```

AdjMatrix DFT

```
-
public String dft(int s) {
    Set set = new Set(m.length);
    return dft(s, s, set);
}
private String dft(int start, int cur, Set set) {
    String result = cur + " ";
    set.union(start, cur);
    for(int i = 0 ; i < m.length ; i++) {
        if(m[cur][i] < Double.POSITIVE_INFINITY && !set.isSameSet(start, i))
            result += dft(start, i, set);
    }
    return result;
}
}
```

Thanks

