CSC373 Worksheet 4 Solution

August 3, 2020

1. • Calculating out-degree

   Let $G = (V, E)$ be a directed graph. Let $[v_1, ..., v_n]$ be a list of vertices in graph G.

   I need to calculate the outdegree of every vertex using adjacency list.

   We know that in addition to counting each v_i in adjacency list where $i = 1, .., n$, we are also counting $|Adj[v_i]|$ edges.

   Since there are $|V| = n$ many vertices, we can write that the total count is $|V| + \sum_{i=1}^{n} |Adj[v_i]| = |V| + |E|$, which is $\mathcal{O}(|V| + |E|)$.

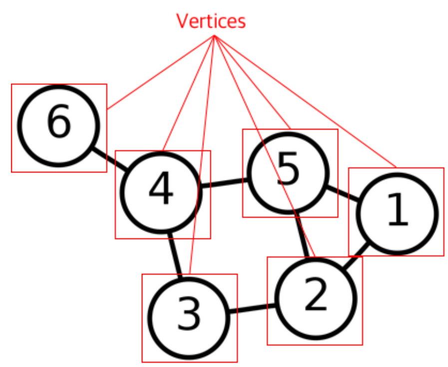
   • Calculating In-degree

   The outdegree of a vertex is indegree of another vertex.

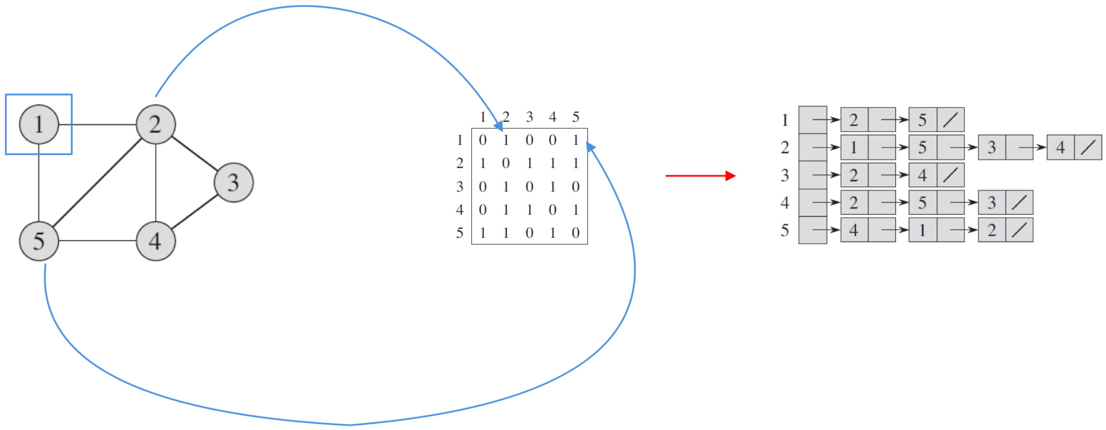
   Using this fact, we can conclude the running time of computing indegree of every vertax is $\mathcal{O}(|V| + |E|)$.

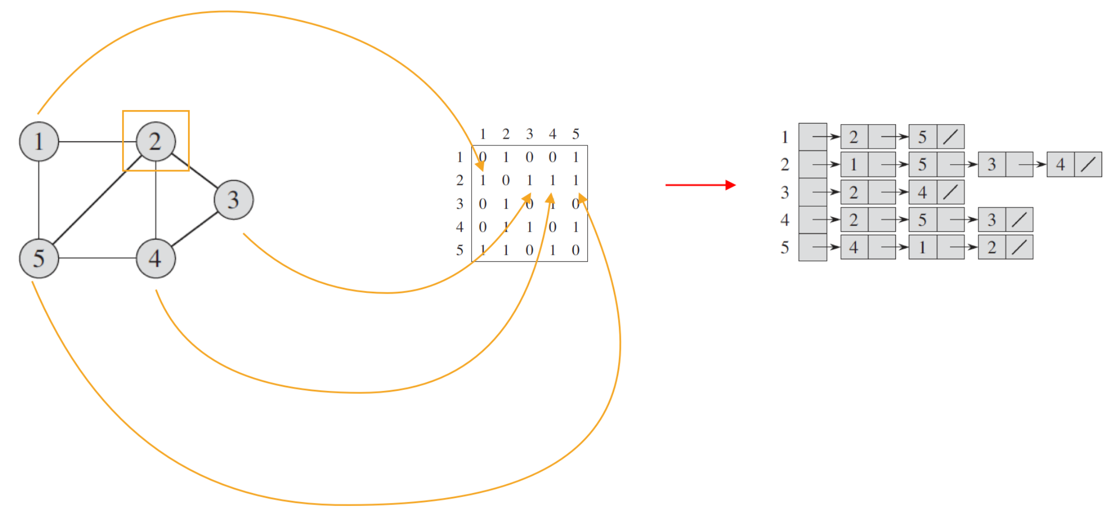
Notes:

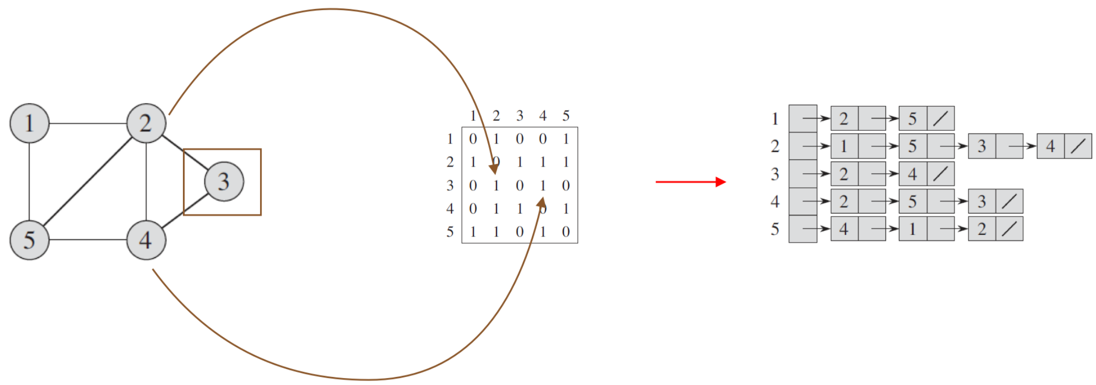
- **Vertex**
  - Is a fundamental unit of which graphs are formed
  - Also means node

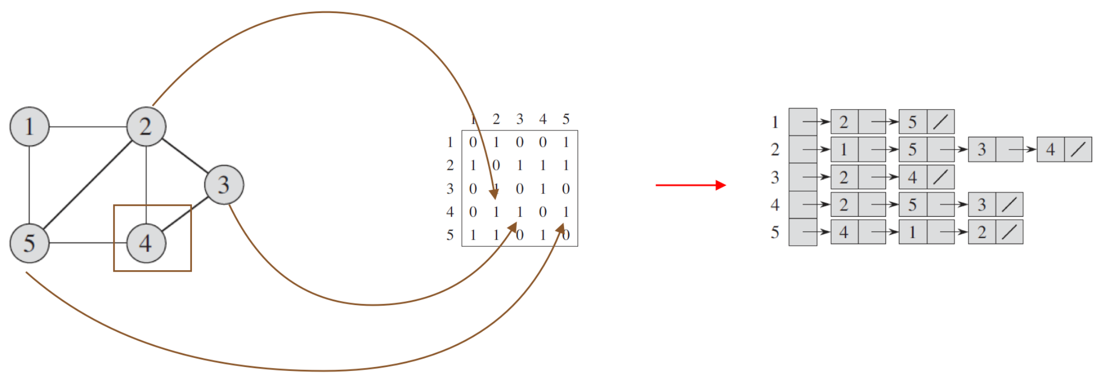


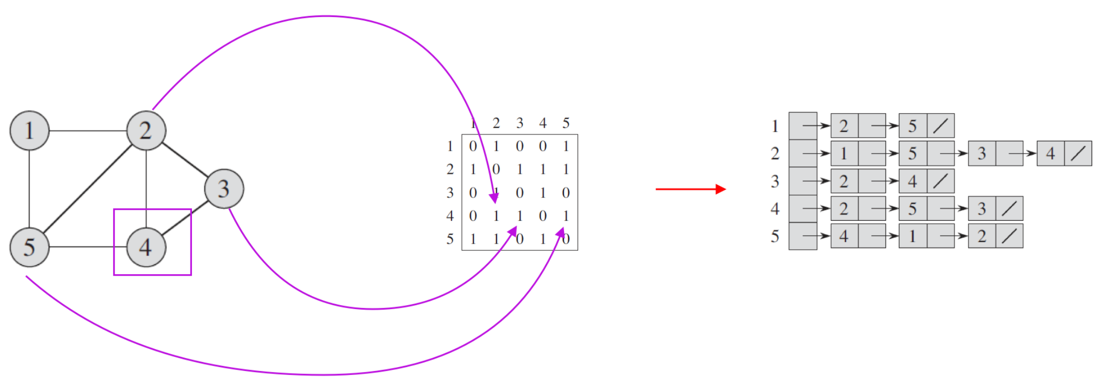
- **Adjacency-list Representation**
  - Associates each vertax in a graph with the collection of its neighbouring vertices or edges
  - Is represented by $Adj[v]$
    * Means all vertices that are neighbour to vertex v
    * In a directed graph, $Adj[v]$ are all out-degree vertices of vertax v
    * $|Adj[v]|$ means the total number of outdegree of vertax v

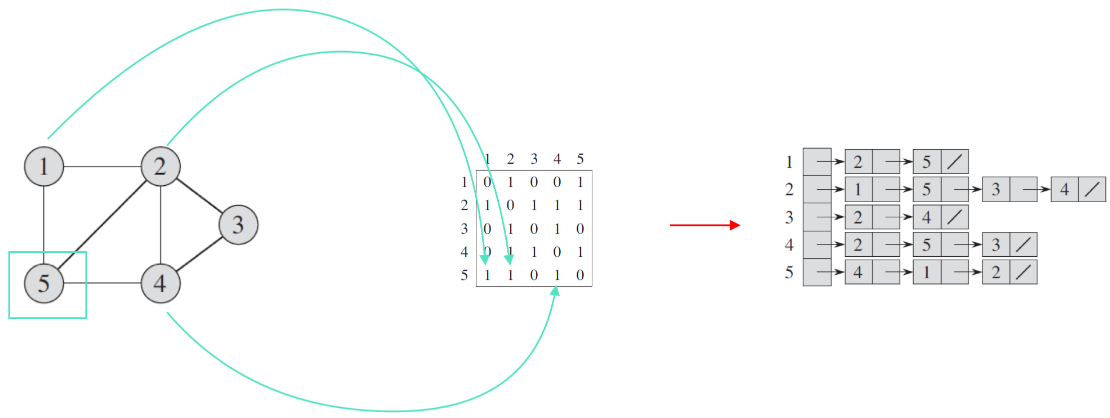




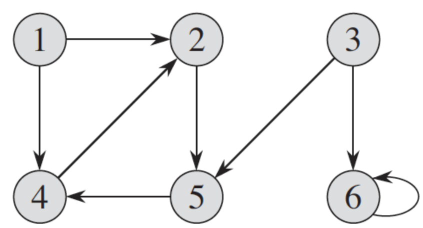

1
2
3
4
5
1 2 3 4 5
1 0 1 0 0 1
2 1 0 1 1 1
3 0 1 0 1 0
4 0 1 1 0 1
5 1 1 0 1 0
1 2 5
2 1 5 3 4
3 2 4
4 2 5 3
5 4 1 2


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

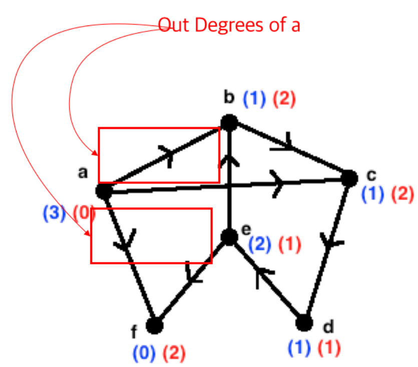

1
2
3
4
5
1 2 3 4 5
1 0 1 0 0 1
2 1 0 1 1 1
3 0 1 0 1 0
4 0 1 1 0 1
5 1 1 0 1 0
1 2 5
2 1 5 3 4
3 2 4
4 2 5 3
5 4 1 2



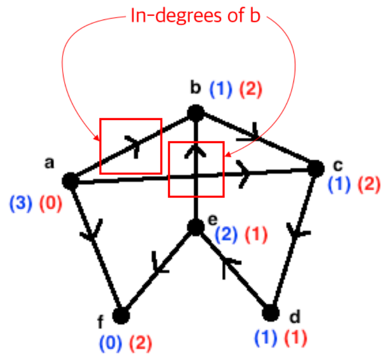
- **Directed graph**
  - Is a graph that is made up of a set of vertices connected by edges, where the edges have a direction associated with them



- **Out-degrees**
  - For a directed graph $G = (V(G), E(G))$ and a vertex $x_1 \in V(G)$, the Out-Degree of x_1 refers to the number of arcs incident from x_1. That is, the number of arcs directed <u>away</u> from the vertex x_1.

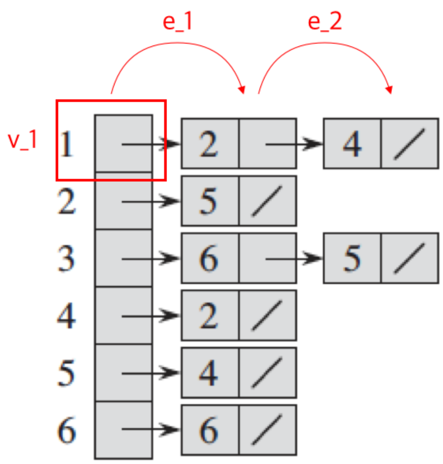


- **In-degrees**
  - For a directed graph $G = (V(G), E(G))$ and a vertex $x_1 \in V(G)$, the In-Degree of x_1 refers to the number of arcs incident to x_1. That is, the number of arcs directed <u>towards</u> the vertex x_1.



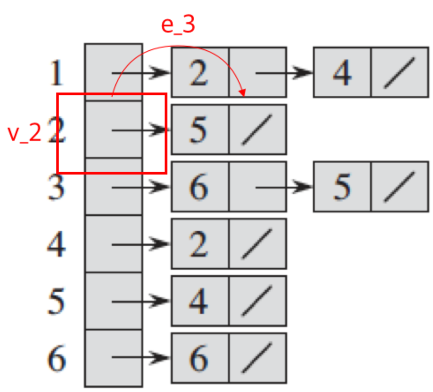
- Computing the outdegree of every vertex using adjacency list

1)



$(v_1) + (e_1 + e_2)$

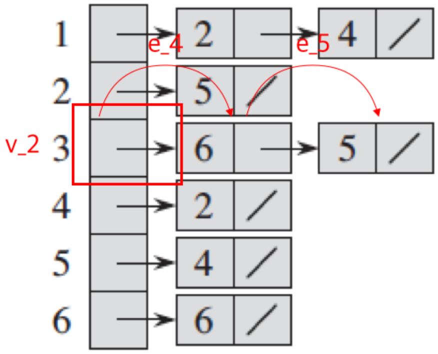
2)



$(v_1 + v_2) +$

$(e_1 + e_2 + e_3)$

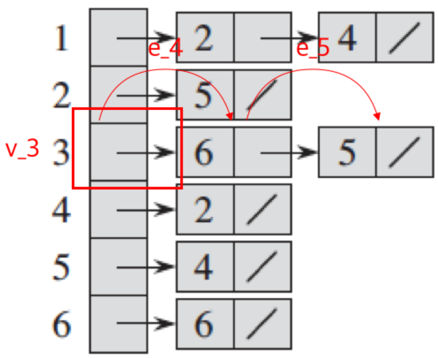
3)



$(v_1 + v_2 + v3)$ +

$(e_1 + e_2 + e_3 + e_4 + e_5)$

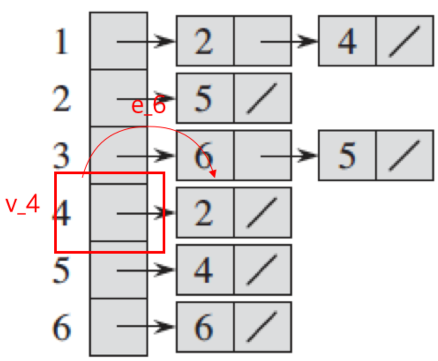
3)



$(v_1 + v_2 + v3)$ +

$(e_1 + e_2 + e_3 + e_4 + e_5)$

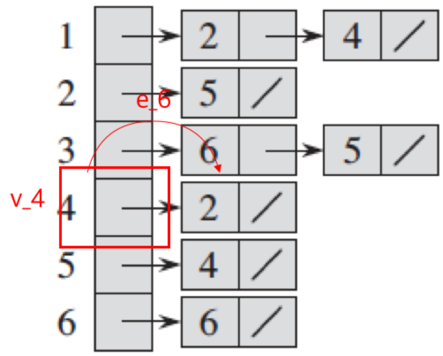
4)



$(v_1 + v_2 + v_3 + v_4)$ +

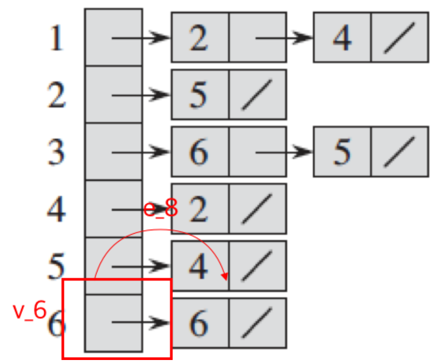
$(e_1 + e_2 + e_3 + e_4 + e_5)$

4)



$(v_1 + v_2 + v_3 + v_4) +$
$(e_1 + e_2 + e_3 + e_4 + e_5)$

6)



$(v_1 + v_2 + v_3 + v_4 + v_5 \_ v_6) +$
$(e_1 + e_2 + e_3 + e_4 + e_5 + e_6 + e_7 + e_8)$

So it has $\mathcal{O}(V + E)$

- Computing the outdegree of every vertex using adjacency list

  The outdegree of a vertex is indegree of another vertex.

  Using this fact, we can conclude the running time of computing indegree of every vertax is $\mathcal{O}(V + E)$.