Joel Turbi

Dr. Zavala

CS312 – Analysis of Algorithms

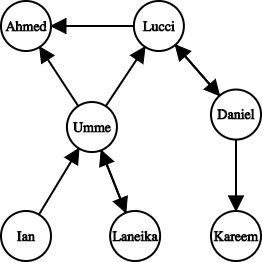
Homework Assignment 9

11/12/2019

# **Reminder**

* The **in-degree** of a vertex is the number of edges leading to that vertex.
* The **out-degree** of a vertex is the number of edges leading away from that vertex.

EXAMPLE:

AhmedUmme

Daniel

In-degree = 2

Out-degree = 0

In-degree = 2

Out-degree = 3

In-degree = 1

Out-degree = 2

**Adjacency-list representation**:



Ahmed



Umme



Daniel

|  |
| --- |
| Ahmed |
| Lucci |
| Umme |
| Ian |
| Laneika |
| Daniel |
| Kareem |

|  |  |  |
| --- | --- | --- |
| Ahmed | Lucci | Laneika |

**Adjacency-matrix representation**:



Umme

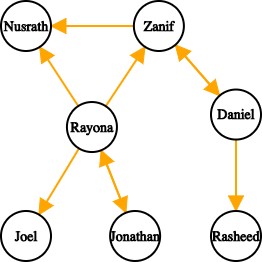


Lucci

Kareem

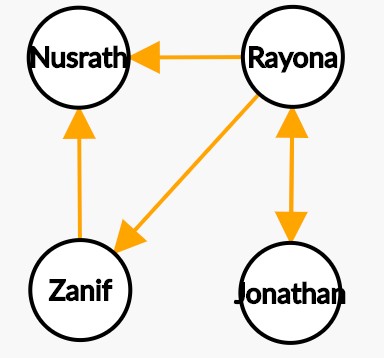
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ahmed | Lucci | Umme | Ian | Laneika | Daniel | Kareem |
| Ahmed | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Lucci | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| Umme | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| Ian | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Laneika | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Daniel | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| Kareem | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Exercises

* 1. Give the in-degree and out-degree of each node in the following graph:

|  |  |  |
| --- | --- | --- |
|  | **In-degree** | **Out-degree** |
| **Nusrath** | 2 | 0 |
| **Rayona** | 1 | 4 |
| **Joel** | 1 | 0 |
| **Jonathan** | 1 | 0 |
| **Zanif** | 2 | 2 |
| **Daniel** | 1 | 1 |
| **Rasheed** | 1 | 0 |

* 1. Give the *adjacency-list* and *adjacency-matrix* representations for the following graph:



**Adjacency-list representation**



Rayona



Nusrath

Jonathan

Zanif



Nusrath

|  |
| --- |
| Nusrath |
| Rayona |
| Zanif |
| Jonathan |

**Adjacency-matrix representation**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nusrath | Rayona | Zanif | Jonathan |
| Nusrath | 0 | 0 | 0 | 0 |
| Rayona | 1 | 0 | 1 | 1 |
| Zanif | 1 | 0 | 0 | 0 |
| Jonathan | 0 | 1 | 0 | 0 |

* 1. Given the ***adjacency-list*** representation of a ***directed graph***:
     1. how would you compute the out-degree of a vertex *v*? (explain the process or algorithm) How long would that take (big-O notation)?
* To compute the out-degree of a vertex one would look at the index of the given node/vertex and count the size of the adjacent list.
* O(1)
  + 1. how would you compute the out-degree of ALL vertices? (explain the process or algorithm) How long would that take (big-O notation)?
* To compute the out-degree of all vertices one would look at all the nodes/vertices and count the total size of the adjacent lists.
* O(V)
  + 1. How would you compute the in-degree of a vertex *v*? (explain the process or algorithm) How long would that take (big-O notation)?
* To compute the in-degree we check in all the adjacent lists; count the number of occurrences of V.
* O(E)
  + 1. How would you compute the in-degree of ALL vertices? (explain the process or algorithm) How long would that take (big-O notation)?
* To compute the in-degree we check in all the adjacent lists; count the number of occurrences for all of the vertices.
* O(V)
  1. Repeat 3 for the ***adjacency matrix*** of a ***directed graph***
     1. how would you compute the out-degree of a vertex *v*? (explain the process or algorithm) How long would that take (big-O notation)?
* To compute the out-degree of a vertex, one would check the given row for V; count the number of 1’s.
* O(V)
  + 1. how would you compute the out-degree of ALL vertices? (explain the process or algorithm) How long would that take (big-O notation)?
* To compute the out-degree of a vertex, one would check the entire row for V; count the number of 1’s.
* O()
  + 1. How would you compute the in-degree of a vertex *v*? (explain the process or algorithm) How long would that take (big-O notation)?
* To compute the in-degree of a vertex, one would check the given row for V; count the number of 1’s.
* O(V)
  + 1. How would you compute the in-degree of ALL vertices? (explain the process or algorithm) How long would that take (big-O notation)?
* To compute the in-degree of a vertex, one would check the entire row for V; count the number of 1’s.
* O()