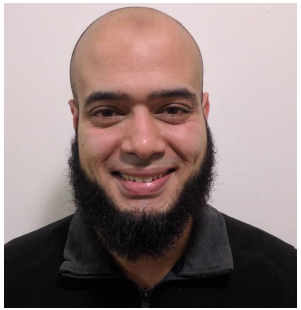
Algorithms

Graph Representation Homework 1

Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Consider during the Graph Sections

- The graphs you are given will always be valid
  - Input vertices are always in the range [0, N-1]
- If graph attributes are not specified assume:
  - **directed unweighted** graph
  - with possibly **multiple edges** and loops
  - implemented using **adjacency list**
  - If the graph is **weighted**: then weights are **integers**
- In **testing**, always consider special graphs
  - Graph of a single node (maybe with a loop)
  - Graph of N nodes and zero edges
  - Graph that is: chain, tree, complete, dense, sparse

Problem #1: Edge List representation

- An edge list representation is a **collection of the graph edges.**
- For example, if we have 10 edges, then we have a vector of 10 edges
- typedef vector<edge> GRAPH;
- Implement a class edge that contains the relevant edges information
  - The edges should be comparable based on the **edge weight**
- void add_edge(GRAPH &graph, int from, int to, int cost)
- void print_adjaceny_matrix(GRAPH &graph)
- Read the edges into the data structure. Print them **ordered** based on cost
- State the space complexity.
- State the time complexity for the possible operations

Problem #2: Adjacency Hashset Representation

- Can we iterate on neighbours efficiently (O(Degree)), while still checking whether an edge exists or not in O(1)? *The best of 2 worlds*?
- Simply, yes. Instead of a list per node, use a **hashset**!
- typedef vector<**unordered_set**<int>> GRAPH;
  - Assume it is unweighted graph
- State the time and space complexity
- State the disadvantages of this representation

Problem #3: Adjacency matrix with multiple edges

- What if we can’t just pick up one edge from the multiple edges?
- Change the adjacency matrix representation somehow to be able to keep the multiple edges
- Print the graph such that the edges of each node are printed first
  - The order of the printed edges per (source) node doesn’t matter
  - See the example
  - The graph edges are directed
  - The graph weights are integer values

Problem #3: Adjacency matrix with multiple edges

```
5 10
0 1 10
1 2 7
0 1 7
2 3 9
2 3 15
3 4 50
0 4 52
0 4 30
2 4 36
4 0 150
From 0 to 1 the cost is 10
From 0 to 1 the cost is 7
From 0 to 4 the cost is 52
From 0 to 4 the cost is 30
From 1 to 2 the cost is 7
From 2 to 3 the cost is 9
From 2 to 3 the cost is 15
From 2 to 4 the cost is 36
From 3 to 4 the cost is 50
From 4 to 0 the cost is 150
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."