COMP37111 - Advanced Algorithms I

Sam Littlefair

October 3, 2018

1 Course Unit Structure

- 10 credit module. Exam worth 75%.
- 2 labs worth 25%.
  1. Coursework A: Due Thursday, 19th October at 12:00 (SSO).
  2. Coursework A: Due Thursday, 23rd November at 12:00 (SSO).

2 Some Basic Graph Algorithms

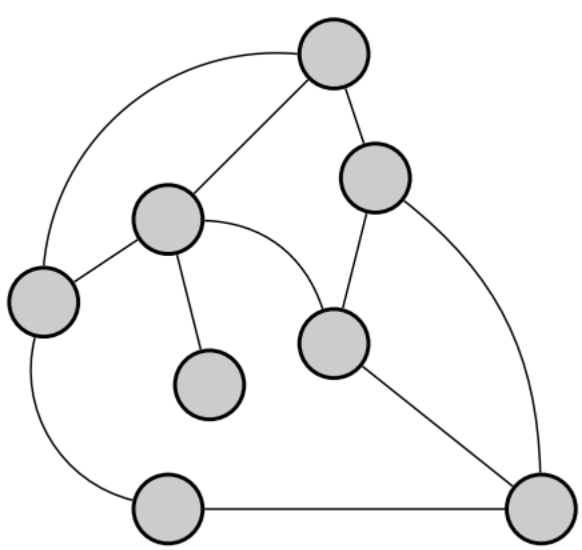
2.1 Graphs and directed graphs

2.1.1 Graphs

Graph: A graph is a pair G = (V, E), where V is a finite set and E a set of subsets of V of cardinality 2. Where V = vertices, E = edges. (Cardinality means number of things in the set.)

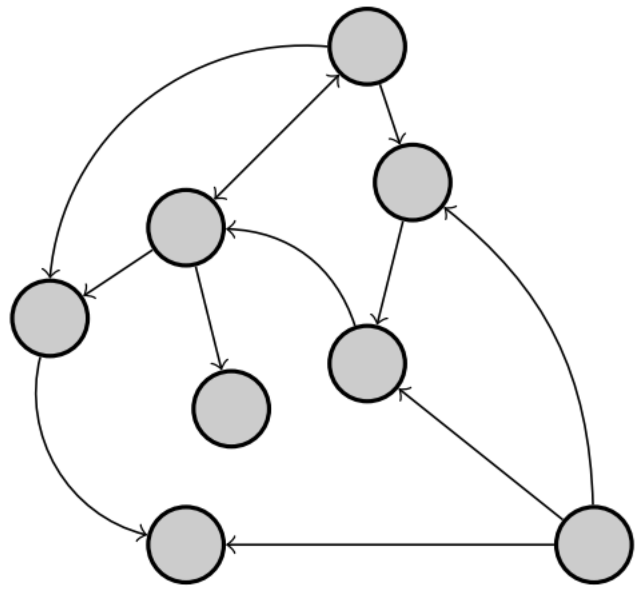
Formal notation for a graph:

- If $\{u, v\} \in E$, we say that u and v are neighbours.
- If $v \in V$, $e \in E$ and $v \in e$, we say v and e are adjacent.



2.1.2 Directed Graphs

Directed graph: A pair G = (V, E), where V is a set and E a set of ordered pairs of distinct elements of V. Where V = vertices, E = edges, the same as a normal graph.



2.1.3 Graph storage

- A directed graph can be stored using adjacency lists:

$$\begin{array}{l} \text{Vertex 1} \rightarrow \ \boxed{2}\ \boxed{3} \\ \text{Vertex 2} \rightarrow \ \boxed{1} \\ \text{Vertex 3} \rightarrow \ \boxed{1}\ \boxed{2}\ \boxed{4} \\ \text{Vertex 4} \rightarrow \ \boxed{1}\ \boxed{3} \end{array}$$

  - From any vertex, the adjacent edges can be accessed efficiently.
  - From any edge, the adjacent vertices can be accessed efficiently.

- An undirected graph can be stored using a symmetric matrix:

$$\begin{array}{cc} & \begin{array}{cccc} V1 & V2 & V3 & V4 \end{array} \\ \begin{array}{c} V1 \\ V2 \\ V3 \\ V4 \end{array} & \begin{pmatrix} * & 0 & 1 & 1 \\ 0 & * & 1 & 1 \\ 0 & 0 & * & 1 \\ 1 & 1 & 0 & * \end{pmatrix} \end{array}$$

  - Wasteful in memory but more convenient.

2.1.4 Graph definitions

Reachable: We say that v is reachable from u if there is a path between them.

Connected: A graph is connected if every node is reachable from every other.

Strongly Connected: A directed graph is strongly connected if every vertex is reachable from every other.

Connected Component: A connected component of a graph is a maximal set of vertices each of which is reachable from any other.

Strongly Connected Component: A strongly connected component of a directed graph is a maximal set of vertices each of which is reachable (in the directed graph sense) from any other.

Cycle: A cycle in a directed graph G is a sequence of vertices $v0, ..., vk = v0(k \geq 2)$ such that, for all i $(0 \leq ik), (vi, vi + 1)$ is an edge. Just a fancy way of saying there is a loop between some vertices.

Cyclic/Acyclic: We call a graph cyclic if it has a cycle, otherwise it's acyclic.

2.1.5 Some problems

Given the definitions above, how can we compute the following functions?

Algorithm 1 Connected Components

1: **Input** A graph G = (V, E).
2: **Return** The connected components of G.

Algorithm 2 Strongly Connected Components

1: **Input** A graph G = (V, E).
2: **Return** The strongly connected components of G.

Algorithm 3 Cyclicity

1: **Input** A directed graph G = (V, E).
2: **Return** True if G is cyclic, False otherwise.