These are some review questions to test your understanding of the material. Some of these questions may appear on an exam.

Graphs

0.1 Define graph, undirected graph, directed graph, and weighted graph, sparse graph.

0.2 Define path in a graph. Define length of a path in a graph.

0.3 Define the following:

   1. Connected, undirected graph.
   2. Strongly connected directed graph.
   3. Weakly connected directed graph.

0.4 Let \( G = (V, E) \) be an undirected graph with \( V \) the set of vertices and \( E \) the set of edges. Let \( v_1, v_2, \ldots, v_p \in V \) be the members of \( V \) and let \( q = |E| \) be the cardinality of \( E \). Prove:

\[
\sum_{i=1}^{p} \text{degree}(v_i) = 2q
\]

0.5 Prove that in any undirected graph, the number of vertices of odd degree is even.

0.6 Write pseudo-code for breadth-first and depth-first traversals of undirected graphs. The code must be complete and must fully describe the operations.

0.7 Describe, in English, any adjacency table graph implementation. How does the implementation differ for directed and undirected graphs?

0.8 Describe, in English, any adjacency list graph implementation. How does the implementation differ for directed and undirected graphs?

0.9 Given a drawing of a directed or of an undirected graph, show its representation as an adjacency matrix.

0.10 Draw the directed graph represented by the adjacency matrix given below. The rows/columns in the matrix correspond to the vertices with labels A,B,C,D,E. A non-zero entry at \([\text{row, col}]\) indicates that the vertex indicated by the row label is adjacent to the vertex indicated by the col label.

\[
\begin{array}{cc}
A & B \\
B & 0 & 1 & 0 & 0 \\
C & 1 & 1 & 0 & 0 \\
D & 0 & 0 & 0 & 1 \\
E & 0 & 0 & 0 & 0 \\
\end{array}
\]
0.11 Given a drawing of a directed or an undirected graph, show its representation as an adjacency list.

0.12 Draw the directed graph represented by the adjacency list given below. This is an “adjacent-to” representation.

\[
v[1] \text{ (Label = A)} \rightarrow 2 \rightarrow 5
\]
\[
v[2] \text{ (Label = B)} \rightarrow 3 \rightarrow 5
\]
\[
v[3] \text{ (Label = C)} \rightarrow 2 \rightarrow 4 \rightarrow 5
\]
\[
v[4] \text{ (Label = D)} \rightarrow 5
\]
\[
v[5] \text{ (Label = E)} \rightarrow \text{empty}
\]

0.13 Discuss the characteristics of the \textit{adjacency table} and \textit{adjacency list} implementations of graphs. Include storage requirements and asymptotic worst-case performance of the operations:

\textbf{Note:} \text{u and v} are \text{vertices in the graph}

- \text{Degree(u)} returns the degree of vertex \text{u} (undirected graphs)
- \text{InDegree(u)} returns the indegree of vertex \text{u} (directed graphs)
- \text{OutDegree(u)} returns the outdegree of vertex \text{u} (directed graphs)
- \text{AdjacentTo(u)} returns a list of the vertices adjacent to \text{u}
- \text{AdjacentFrom(u)} returns a list of the vertices adjacent from \text{u}
- \text{Connected(u,v)} returns true if there is an edge between \text{vertices u and v}, returns false otherwise

0.14 Consider the directed graph represented by the following adjacency matrix (an adjacent-to representation):

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

List the depth-first and breadth-first traversals of the graph beginning at vertex A. Repeat for vertex B. (Whenever a new vertex is to be visited and there is more than one possibility, use the vertex labelled with the letter than comes first in the alphabet.)

0.15 Define \textit{directed acyclic graph}.

0.16 Define \textit{topological ordering} of a directed acyclic graph.

0.17 Given a drawing of a graph, find all cycles.

0.18 Given a drawing of a directed acyclic graph, write the labels of its vertices in topological order. Explain how you obtained the ordering.

0.19 Given a drawing of a directed graph along with the “discovery” and “finish” times of its vertices after a depth-first search, write the labels of the graph in topological order. Explain how you obtained the ordering.

0.20 Given a drawing of a directed graph along with the “discovery” and “finish” times of its vertices after a depth-first search, identify the type of each edge (tree, back, forward, or cross). The following test is relevant, with \(d[v]\) the discovery time of vertex \(v\) and \(f[v]\) its finish time:

2
if ( (d[v1] < d[v2]) && (f[v1] > f[v2]) )
    (v1,v2) is a tree edge
else if (d[v1] > d[v2] && f[v1] < f[v2])
    (v1,v2) is a back edge
else if (d[v1] > d[v2] && f[v1] > f[v2])
    (v1,v2) is a cross edge
    (v1,v2) is a forward edge