Graphs 1

Monday, March 23, 2020
10:58 AM

Today:
- Graphs terminology
- Modelling problems
- Representing graphs
- Graph traversals/graph search

\[ G = (V, E) \]

\[ V: \text{ set of vertices,} \]
\[ E \subseteq V^2 \]
\[ E = \{(v_1, v_2) \ldots \} \]

"there is an edge between \( v_1 \) and \( v_2 \) in \( G \)"

- Other features:
  - Attributes about the nodes or about the edges
    - Aka label
  - Directed or undirected

Undirected graph:
\[ (v_1, v_2) \in E \iff (v_2, v_1) \in E \]

- Directed graph
  - OK to have edge between \( (v_1, v_2) \) and \( \text{'multigraph'} (v_2, v_1) \)

- Simple undirected graph:
  - At most one edge \( (v_1, v_2) \)
  - No self loops

- Trees

Model a problem with a graph

<table>
<thead>
<tr>
<th>Example</th>
<th>What are the nodes?</th>
<th>What are the edges?</th>
<th>What are the attributes?</th>
<th>Directed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nodes</td>
<td>Transitions</td>
<td>Attributes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Abstract questions about graphs

I is there a path between \( v_1 \) and \( v_2 \)

\[
G_1 = \begin{pmatrix}
2 & 0 & 3 \\
4 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix}
\]

E.g. no path b/w 1 and 4

Source/Start \( \rightarrow \) Sink/End

Desc: Path between \( v_0 \) and \( v_k \) is a sequence of nodes \((v_0, v_1, \ldots, v_k)\) s.t.

V: all possible game states

\( E: (v_1, v_2) \) iff

\( v_2 \) is the result of applying a valid chess move.

\( \checkmark \) directed \( \rightarrow \) not always reversible.
For all \( i \in \mathbb{N} \), if \( (v_i, v_{i+1}) \in E \) related: What's the shortest path?

If there are weights on edges, may be use smallest or largest weight path

\[ A \rightarrow B \text{ of length } 1 \]
\[ A \rightarrow B \text{ shortest weighted path is } 3 \]

Connectedness

\[ G = (V, E) \]

Defn: unconnected graph:

\[ \exists v_i, v_j \text{ s.t. no path from } v_i \rightarrow v_j \]

Count the number of connected components

Independent sets: set of nodes \( \{v_1, ..., v_n\} \) s.t. no pair has an edge b/w them

Representing graphs

- Adjacency matrix: \( E \leq V^2 \)

\[
\begin{array}{cccccc}
 & 1 & 2 & 3 & 4 & 5 \\
1 & 0 & 1 & 1 & 0 & 0 \\
2 & 1 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

degree(3)?
- List of lists (array of arrays)
  
  For each node, store a list of neighbors:
  
  1: [2, 3]
  2: [1, 3]
  3: [2, 1]
  4: [3]
  5: [4]
  6: []

  Metrics: \( G = (V, E) \)
  
  \( n = |V| \) # of nodes
  \( m = |E| \) # of edges

  Constraints: \( m \leq n^2 \) (for a simple graph)

  \( O(n + m) \) is a smaller bound \( O(n^2) \)

\[
\begin{array}{|c|c|c|}
\hline
\text{Storage} & \text{Run-time for \( deg() \)} & \text{Run-time for \( deg() \)}
\hline
\text{adj. mat} & n^2 & O(n^2) & O(n)
\hline
\text{list of lists} & O(n + m) & O(n + m)
\hline
\text{array of arrays} & O(n + m) & O(1)
\hline
\end{array}
\]

degree \( (v) \): # of neighbors

---

Traversals:

Given \( V \), find all nodes in \( G_{\in}(V) \)

\( G_{\in}(V) \): \( \exists u, v \in V \) \text{Path } U \rightarrow V \)

\( \text{Visited} = \left[ \begin{array}{c}
\text{False} \\
\text{False} \\
\text{False} \\
\text{False} \\
\text{False} \\
\text{False}
\end{array} \right] \) // size n array
- \textbf{Explore}(v):
  // Start from \textit{v},
  // every node reachable from \textit{v} is \textit{visited}!
  if \textit{v} not visited: mark \textit{v}
  for \textit{neighbor} \textit{w} \in \text{neighbors}(\textit{v}):
    if visited[\textit{w}] = False:
      set visited[\textit{w}] = True
      \textbf{Explore}(\textit{w})