Graphs

Graph: represents things and relationships between them

- nodes/vertices: people
- edges: relations

Path: sequence of vertices w/ edges between $J6, LH, FM, FDR$

- Simple path: no repeats

- Cycle: starts/ends at same place $J6, MS, KK, J6$

- Simple cycle: only repeat at beginning and end

For any two vertices, is there a path between of length $\leq 6$
int foo(int n, int c)
{
    if (n == c)
    {
        return 0;
    }

    int i = 1;
    while (i < n)
    {
        if (i % c == 3)
        {
            if (n % 2 == 1)
            {
                return 0;
            }
        }
        i++;
    }
}
verts: teams  edge $u \rightarrow v$: $u$ lost to $v$

vertices

edges

Can we order teams so edges go in same direction?

What ordering minimizes edges in wrong direction?

Updates

MIN-FEEDBACK-ARC-SET

NP-complete
vertices: cities
dges: roads between
weights: travel times

what path, start -> end
has min. total weight

vertices: intersections
dges: segments of roads
(w/ direction to represent one-way)
Graph Representation

Adjacency Matrix

\[
\begin{array}{cccccc}
  & Y & C1 & D2 & Pr & H \\
Y & F & F & F & F & F \\
C1 & T & F & F & F & F \\
D2 & T & F & F & F & F \\
Pr & T & F & F & F & T \\
H & F & T & T & F & F \\
\end{array}
\]

Adjacency List

- List of lists s.t. `Adj[v]` is list of `u` s.t. `v \rightarrow u` is an edge

Y :
Col : Y D
D : Y
Pr : Y D H
H : Y Col D

Map `has EDGE`:
- `H2 Col`
<table>
<thead>
<tr>
<th>Adj Matrix</th>
<th>Adj List</th>
<th>( n = # ) vertices</th>
<th>( m = # ) edges</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>space</em></td>
<td>( O(n^2) )</td>
<td>( O(n^2) )</td>
<td>( 0 \leq m \leq n^2 )</td>
</tr>
<tr>
<td><em>has_edge</em></td>
<td><em>O(1)</em></td>
<td>( O(n) ) (seq. search)</td>
<td>( m &lt; n^2 ) (ex. ( m = O(n) ))</td>
</tr>
<tr>
<td><em>add_edge</em></td>
<td><em>O(1)</em></td>
<td>( O(n) ) (has_edge to avoid duplicates)</td>
<td></td>
</tr>
<tr>
<td><em>for_each_neighbor</em></td>
<td>( O(n) )</td>
<td>( O(n) ) (degree(n))</td>
<td></td>
</tr>
<tr>
<td><em>for each vertex v</em></td>
<td>( O(n^2) )</td>
<td>( O(n^2) ) worst case</td>
<td>( \sum_v \sum_{\text{deg}(v)} 1 )</td>
</tr>
<tr>
<td><em>for each neighbor of v</em></td>
<td>( O(n^2) )</td>
<td>( O(n^2) )</td>
<td>( \sum_v \sum_{\text{deg}(v)} 1 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>better for sparse graphs</td>
<td>( = n + m )</td>
</tr>
</tbody>
</table>