CSE 421 Lecture 4

1 Testing Bipartiteness

Theorem 1. *We can test if a graph with m edges and n vertices is bipartite in $O(m+n)$ time.*

Proof:
Since we can test bipartiteness separately in each connected component, we assume the graph is connected.

Algorithm:

- Run BFS starting at any vertex.
- If there is any edge in the graph that joins two vertices of the same layer,
  - Output non-bipartite.
- Else
  - Output bipartite.

.

Runtime:
$O(m+n)$. The bottleneck is BFS.

Correctness:
Case 1: The algorithm outputs "bipartite".
Color nodes on even layer blue, odd layer red.
Since no edge on the same layer (assumption) and all edges join nodes on adjacent layers (property of BFS), the 2 color we gives is valid.
Case 2: The algorithm outputs non-bipartite.
There is an edge xy connecting two nodes on the same layer.
Let z be their lowest common ancestor.
Note that the path $x \to y \to z \to x$ forms an odd cycle.
However, there is no way to have 2 coloring for odd cycle.
Hence, the whole graph does not have 2 coloring.

Corollary 2. *A graph is bipartite if and only if it has no odd cycle.*

2 Non-Tree Edge in DFS

Lemma 3. *For every undirected edge $\{x, y\}$, then one of x or y is an ancestor of the other in the tree.*

Proof:
Suppose x is visited first.
Therefore, DFS(x) was called before DFS(y).
Case 1: $\{x, y\}$ is in the DFS tree and that means y is a child of x.
Case 2: y was visited when the edge $\{x, y\}$ was examined during DFS(x). In this case, y is a descendant of x.

3 Topological order

Theorem 4. *G has a topological order if and only if G is a DAG.*

We split the proof into few parts. First we prove DAG is necessary.

Lemma 5. *If G has a topological order, then G is a DAG.*

Proof:
Our goal is to prove G does not has a directed cycle.
To prove by contradiction, assume there is a directed cycle C.
We name the vertices of G by some topological order $1, 2, 3, \cdots, n$
Let i be the lowest-indexed vertex in C. Let j be the vertex before i.
Then, (j, i) is a directed edge with $i < j$. This contradicts to the topological order.

To prove any DAG has a topological order, we first need to find some starting vertex (aka course without prerequisite)

Lemma 6. *If G is a DAG, G has a source (vertex with no incoming edges).*

Proof: (by contradiction)
Suppose G is a DAG with no source.
Pick any vertex v.
Since v is not a source, there is an income edge (u, v) and we can follow the edge backward to u.
Repeat walking backward until we visit a vertex w twice.
Let the walk be $w \to \cdots \to w \to \cdots \to v$.
The part $w \to \cdots \to w$ is a directed cycle. (Contradiction)

Lemma 7. *If G is a DAG, G has a topological order*

Proof: (by induction on n)
Statement $P(n)$**:** Every DAG with n vertices has a topological ordering
Base case $n = 1$: True
Induction:
Given a DAG G with n vertices. Let v be a source.
Note that $G - \{v\}$ is a DAG.
By hypothesis, $G - \{v\}$ has a topological ordering π.
Hence, (v, π) is a topological order for G (because v has no incoming edge).