Worksheet on Directed Graphs

Benjamin Cosman, Patrick Lin and Mahesh Viswanathan

Fall 2020

TAKE-AWAYS

- A *directed graph* (or *digraph*) is a set of *vertices* along with a set of directed *edges* that each point from one vertex to another.
- Informally, a digraph is just a bunch of dots and arrows, which can be a useful visualization of a relation $R \subseteq A \times A$.
- A *walk* is an alternating list of vertices and the edges that connect them (usually described using just the vertices or just the edges). A *path* is a walk that does not repeat vertices. A *closed walk* is a walk that ends where it begins. A *cycle* is a positive-length closed walk that does not repeat vertices (other than the end being a repeat of the start).
- $dist(u, v)$ (short for *distance*) is the length of the shortest path from u to v.
- Two walks can be *merged* into one walk $w_1 \widehat{x} w_2$ if w_1 ends at x and w_2 starts at x.
- A *DAG* (directed *acyclic* graph) is a digraph with no cycles. A *cyclic* digraph has one or more cycles.
- A *topological sort* of a digraph is a list of all its vertices such that for each edge (u, v), u comes before v in the list. On this week's homework you will prove that a digraph has a topological sort if and only if it is a DAG.
- A relation R is *reflexive* if $\forall x(xRx)$.
- A relation R is *symmetric* if $\forall x \forall y(xRy \rightarrow yRx)$.
- A relation R is *transitive* if $\forall x \forall y \forall z((xRy \wedge yRz) \rightarrow xRz)$.
- A relation is an *equivalence relation* if it is reflexive, symmetric, and transitive.

Problem 1. Consider a digraph G where $V(G)$ is the set of all currently-living humans. For each part below, list two possible definitions for $E(G)$ such that G has (or probably has) the given property. [1] You do not need to justify your answers.

[1] For example, one possible answer to part (b) is $E(G) = \{(x, y) | x$ is a child of $y\}$. There will be no cycles in that graph (imagine if there were - which member of the cycle was born first?), and there is also no single path traveling through all the vertices (for one thing, for any pair of siblings, there will be no path that contains both of them in either order).

a) G is acyclic, and there is some path that travels through every vertex.

b) G is acyclic, and no path travels through every vertex.

c) G is cyclic, and for every edge between two people there is a matching edge going in the other direction.

d) G is cyclic, and some edges do not have matching edges going in the other direction.

Problem 2. Prove or disprove each statement.

a) There exists a graph with at least 3 vertices where every possible closed walk is a cycle.

b) For every graph, if a, b, c are vertices, w is a shortest path from a to b, and p is a shortest path from b to c, then $|w\frown p| = dist(a, c)$.

Problem 3. Give a binary relation $R \subseteq \{1, 2, 3\} \times \{1, 2, 3\}$ such that $(1, 2) \in R$ and R is symmetric and transitive, but R is not an equivalence relation.

Problem 4. Find a digraph G with 3 vertices such that $E(G) = E(G^4)$.[2] Then find two more such graphs; try to make your 3 answers as different from each other as you can.

[2] Recall: $(u, v) \in E(G^4)$ iff there is a walk of length exactly 4 from u to v in G

Problem 5. ProofBlocks is the tool on PrairieLearn where you re-order tiles to construct a proof, designed by our own TA Seth Poulsen. Fill in the blanks below to describe how ProofBlocks uses directed graphs.

There is a vertex for each ______. There is an edge from u to v if ______. A correct proof that uses all the blocks corresponds to a ______ of the graph, because ______.

Problem 6. Imagine an ant crawling around on a DAG G. The ant can only see the edges coming in and out of its current vertex, and can crawl along them (in either direction). It has no memory, and only enough energy to crawl along $2|V(G)|$ edges total.

a) If the ant starts at a random vertex, describe an algorithm it can use that will always let it find a vertex with indegree 0.[3]

[3] The indegree of a vertex v refers to the number of incoming edges, i.e. the number of vertices a where (a, v) is an edge in the graph

b) Prove your algorithm works. *(Convince the reader that it does not have any steps that are impossible to follow, and that it will end before the ant runs out of energy.)*

c) Would your algorithm still work if the graph had a cycle? (could any algorithm?)