Solutions to CS 70 Challenge Problems: Graphs, Trees, and Hypercubes

Other worksheets and solutions at https://alextseng.net/teaching/cs70/

Alex Tseng

---

1 Trees

(a) Prove or give a counterexample: a tree must have some vertex of degree 1.
This is true. First we realize that a tree cannot have any vertices of degree 0 (isolated vertices) because a tree must be connected.
Then we are equivalently showing that any tree must have some vertex with degree less than 2 (since the degree cannot be 0, it must be 1). We utilize a proof by contradiction.
Assume for contradiction that all vertices in some tree with n vertices have degree at least 2. Whenever an edge touches a vertex, we call that an "incidence". If some vertex v has degree $d(v)$, then it contributes $d(v)$ incidences to the tree. The sum of all incidences $\sum_{v \in T} d(v) \geq 2n$, since every degree is at least 2. Additionally, we know that every edge contributes exactly 2 incidences. This would imply that there are at least n edges (there are at least $2n$ incidences). Contradiction—a tree with n vertices has exactly $n-1$ edges. Therefore, a tree must have some vertex with degree less than 2, namely 1.

2 Hypercubes

(a) We define the following function:
$s(0) = 1$
$s(n) = 2^{n-1}(s(n-1))^2$
Prove that an n-dimensional hypercube has at least $s(n)$ spanning trees.
Note that a spanning tree of graph G is a subgraph that is a tree, and that connects all vertices of G.
Given the recursive nature of the function s, and the recursive nature of hypercubes, it should be obvious that induction is the way to go.
Base Case: For a 0-dimensional hypercube (a single vertex), there is exactly 1 spanning tree—the node itself.
Inductive Hypothesis: Assume that for an $(n-1)$-dimensional hypercube, there are at least $s(n-1)$ spanning trees.
Inductive Step: We start with a hypercube H of n dimensions. We break it down arbitrarily into two hypercubes H_1 and H_2, each of $n-1$ dimensions. By the Inductive Hypothesis, each one has at least $s(n-1)$ spanning trees. Note that if we have a spanning tree in H_2 and a spanning tree in H_1, they collectively connect all the vertices in each hypercube. Then to construct a spanning tree for H, we just need to connect any two vertices in the spanning trees of H_1 and H_2. Any spanning tree of H is distinctly defined by which spanning tree we pick from H_1, which spanning tree we pick from H_2, and what edge we use to connect them. First we pick a pair of spanning trees. There are $(s(n-1))^2$ such pairs. For each pair, we can connect them with any of the edges in H that goes between H_1 and H_2. There are 2^{n-1} such edges. Then we have just constructed at least $2^{n-1}(s(n-1))^2$ spanning trees in H that are all different.
Thus, by induction, an n-dimensional hypercube has at least $s(n)$ spanning trees.

(b) Is the bound in the question above a tight bound? That is, is it possible that an n-dimensional hypercube has more than $s(n)$ spanning trees?
No, $s(n)$ is not a tight bound. The reason is that there are more ways to construct a spanning tree than to split the graph in half and connect spanning trees from each half. For example, we can take a path that starts in H_1, goes to H_2, and then ends in H_1, covering all vertices. A more concrete example would be a 2-dimensional hypercube (a square). There are 4 possible spanning trees, but $s(2) = 2$.

3 General Graphs

(a) Let G be an undirected graph with $n \geq 2$ vertices. Every vertex in G has an even degree, except for the vertices u and v, which have odd degrees. Prove or give a counterexample: there must be a path between u and v.
This is true. We assume for contradiction that there is no path between u and v. Then u and v must be in separate components with no path in between. Then each component has a sum of degrees that is odd, since each component has exactly one vertex with odd degree. Contradiciton—the sum of the degrees of any connected component of an undirected graph is 2 times the number of edges in that component, so it must be even.

(b) For some graph G, the graph $\overline{G}$ (G-complement) has all of the same vertices, but has the opposite edges. That is, if an edge (u, v) exists in G, then it does not exist in $\overline{G}$, and if the edge does not exist in G, then it does exist in $\overline{G}$. Prove that for any graph G, either G is connected or $\overline{G}$ is connected. You may assume that the graph is undirected.
If G is connected, then we are done. So let us assume that G is not connected. Then we wish to show that $\overline{G}$ is connected. $\overline{G}$ is connected if every pair of vertices in $\overline{G}$ have some path between them. Without loss of generality, let us consider any two distinct vertices u, v in $\overline{G}$.
Now we utilize a proof by cases to show that u and v must have a path in $\overline{G}$.
If u and v were in different connected components in G (equivalently, there was no path from u to v in G), then there was no edge (u, v) in G. Then in $\overline{G}$, there must be the edge (u, v), so there is a path between u and v.
In the case that u and v were connected in G, they were in the same component. But since G is disconnected, there must be some other non-empty component that is not connected to u or v. Let any vertex in that component be w. Since w is not connected to u or v in G, the edges (u, w) and (v, w) must exist in $\overline{G}$. Then u and v are connected in $\overline{G}$ through the vertex w.
Therefore, if G is disconnected, any pair of vertices are connected in $\overline{G}$, and therefore $\overline{G}$ is connected.

(c) *Challenge* Prove that a *complete* undirected graph on n vertices (assume n is even) can be partitioned into $\frac{n}{2}$ edge-disjoint spanning trees. That is, we can find $\frac{n}{2}$ different spanning trees where no edge is ever used twice.
We utilize a proof by induction on the n, the number of nodes in the graph.
<u>Base Case</u>: For a 2-node complete graph, there is only one spanning tree.
<u>Inductive Hypothesis</u>: Assume that for an n-node complete graph, there are $\frac{n}{2}$ edge-disjoint spanning trees.
<u>Inductive Step</u>: Consider a complete graph of $n + 2$ nodes. We arbitrarily remove any two nodes and their incident edges. Let these nodes be a and b. The remaining graph is still complete and has n nodes. Let these nodes in the remaining graph be $v_1, ..., v_n$. By the Inductive Hypothesis, there are $\frac{n}{2}$ edge-disjoint spanning trees in this subgraph. Let these trees be $T_1, ..., T_{\frac{n}{2}}$.
Now we add back nodes a and b (and their edges). We need to extend each tree to also include a and b for them to be spanning. Note that since the graph is complete, both a and b have edges to each v_i, and there is also the edge (a, b).
To each tree T_i, we add the edges (a, v_i) and $(b, v_{\frac{n}{2}+i})$. For example, to extend T_1, we add the edges (a, v_1) and $(b, v_{\frac{n}{2}+1})$, and to extend $T_{\frac{n}{2}}$, we add the edges $(a, v_{\frac{n}{2}})$ and (b, v_n). Thus, we extend each T_i to create $\frac{n}{2}$ spanning trees of $n+2$ vertices. Since each T_i was edge-disjoint from the others, and all the edges we added were different for each T_i, the resulting spanning trees are also edge-disjoint.
We construct one more spanning tree. We use the edges $(a, v_{\frac{n}{2}+1})$ through (a, v_n), and the edges (b, v_1) through $(b, v_{\frac{n}{2}})$, as well as the edge (a, b). These edges have not been used yet, and they connect all the vertices to form one last spanning tree. Thus, we now have $\frac{n}{2} + 1 = \frac{n+2}{2}$ edge-disjoint spanning trees.
Therefore by induction, a complete graph with n nodes (n is even) will have $\frac{n}{2}$ edge-disjoint spanning trees.

(d) *Challenge* Given a tournament H of $n \geq 3$ vertices, show that there exists some node that is reachable from every other node on a path that is at most length 2. That is, we can identify some vertex x in H where if we start on any other vertex in H, we can travel to x by traversing at most 2 edges.

Recall that a tournament is a directed graph where every pair of nodes u and v are connected by either an edge (u, v) or (v, u), but not both.

We utilize a proof by induction on n, the number of nodes. In this proof, we will use the term "nexus" to describe the vertex x, which can be reached from every other node in no more than 2 edges.

Base Case: Consider a tournament of 3 vertices. Since there are only 3 nodes, we can identify a single node to be the nexus, which *must* be at most 2 edges away from the other nodes. If this is hard to visualize, try drawing out all the possible 3-node tournaments (there are only 2).

Inductive Hypothesis: Assume that for a tournament of n vertices, there exists a nexus.

Inductive Step: Consider a tournament of $n + 1$ vertices. We choose any vertex to remove. Let this vertex be v. The resulting graph is still a tournament, but with n nodes. By the Inductive Hypothesis, we can find a nexus. Let this nexus be x. In this smaller tournament, we can split the vertices into three groups: x itself is in one group, the "1-nodes" are the nodes that have a direct edge to x, and the "2-nodes" consists of all the other nodes that do not have a direct edge to x, but can reach x through a node in the "1-nodes" group.

We add back v to get our $n + 1$-node tournament. In the first case, if v has an edge to x, or an edge to any node in the "1-node" group, then v has a path of at most 2 edges to x, maintaining x as the nexus. In the second case, x has an edge to v, and every node in the "1-nodes" group has an edge to v. Then all of these nodes can reach v in exactly 1 edge. All of the nodes in the "2-nodes" group have an edge to some node in the "1-nodes" group, and so they have paths to reach v in exactly 2 edges. Thus, v is a nexus.

Thus, there is always a nexus, either x or v.

Therefore by induction, any tournament of at least 3 nodes has a nexus node.