

Problem Set 5

Due: Monday, October 7

Problem 1. [20 points]

In lecture we discussed the notion of pairing up boys and girls by finding a *minimum weight matching* on the bipartite graph, G , where the weight of an edge $b—g$ is the sum of the rank of the g on b 's list plus the rank of b on g 's list. The minimum weight matching of G is the matching that produces the lowest sum of weights on the edges of G .

Ex. If boy b_1 has a ranking of girls g_1, g_2, g_3, g_4 , and girl g_1 has a ranking of boys b_2, b_3, b_4, b_1 , then the weight of the edge $b_1—g_1$ will be $1 + 4 = 5$, since g_1 is ranked first and b_1 is ranked fourth.

(a) [10 pts] Prove that the minimum weight matching is not always a stable matching by providing a counterexample.

(Hint: There is a counterexample with 3 boys and 3 girls.)

(b) [10 pts] The minimum weight matching minimizes the sum over *all* possible matchings. Instead, consider a greedy algorithm that recursively matches the minimum weight edge over all unmatched nodes. Prove that this matching is also not always a stable matching by providing a counterexample.

Ex. If all boys have the same ranking of girls g_1, g_2, g_3, g_4 , and all girls have the same ranking of boys b_1, b_2, b_3, b_4 , then $b_1—g_1$ will be matched first (weight= 2), $b_2—g_2$ will be matched second (weight= 4), $b_3—g_3$ will be matched third (weight= 6), and $b_4—g_4$ will be matched last (weight= 8).

Note two things:

- Suppose that, in the case of a tie for minimum weight edge, the algorithm matches all of the tied edges. If this creates a conflict (for example, if $g_1—b_1$ and $g_1—b_2$ both have weight= 3), then the algorithm just matches one of the pairs randomly (for example, $g_1—b_1$). Choose a counterexample with no such conflict.
- The algorithm does *not* recalculate the weights after each recursion, so the weight of an edge does not change, even as higher ranked preferences become unavailable. There is also a counterexample to stability for such an algorithm that does reweight the edges, but it's much more difficult to find!

(Hint: There is a counterexample with 4 boys and 4 girls.)

Problem 2. [15 points]

(a) [5 pts] Suppose that G is a simple, connected graph on n nodes. Show that G has exactly $n - 1$ edges *iff* G is a tree.

(b) [10 pts] Prove by induction that any connected graph has a spanning tree.

Problem 3. [20 points] An n -node graph is said to be *tangled* if there is an edge leaving every set of $\lceil \frac{n}{3} \rceil$ or fewer vertices. As a special case, the graph consisting of a single node is considered tangled. (Recall that the notation $\lceil x \rceil$ refers to the smallest integer greater than or equal to x .)

(a) [7 pts] Find the error in the proof of the following claim.

Claim. Every non-empty, tangled graph is connected.

Proof. The proof is by strong induction on the number of vertices in the graph. Let $P(n)$ be the proposition that if an n -node graph is tangled, then it is connected. In the base case, $P(1)$ is true because the graph consisting of a single node is defined to be tangled and is trivially connected.

In the inductive step, for $n \geq 1$ assume $P(1), \dots, P(n)$ to prove $P(n + 1)$. That is, we want to prove that if an $(n + 1)$ -node graph is tangled, then it is connected. Let G be a tangled, $(n + 1)$ -node graph. Arbitrarily partition G into two pieces so that the first piece contains exactly $\lceil \frac{n}{3} \rceil$ vertices, and the second piece contains all remaining vertices. Note that since $n \geq 1$, the graph G has a least two vertices, and so both pieces contain at least one vertex. By induction, each of these two pieces is connected. Since the graph G is tangled, there is an edge leaving the first piece, joining it to the second piece. Therefore, the entire graph is connected. This shows that $P(1), \dots, P(n)$ imply $P(n + 1)$, and the claim is proved by strong induction. \square

(b) [5 pts] Draw a tangled graph that is not connected.

(c) [8 pts] An n -node graph is said to be *mangled* if there is an edge leaving every set of $\lceil \frac{n}{2} \rceil$ or fewer vertices. Again, as a special case, the graph consisting of a single node is considered mangled. Prove the following claim. *Hint: Prove by contradiction.*

Claim. Every non-empty, mangled graph is connected.

Problem 4. [10 points] Let the nodes in a tournament be ranked according to their outdegrees, that is, node u is ranked less than or equal to v iff $\text{outdegree}(u) \leq \text{outdegree}(v)$. Prove that the sum of the outdegrees of the i lowest ranked nodes is at least $i(i-1)/2$.

Problem 5. [10 points] A directed graph is *symmetric* if, whenever $x \rightarrow y$ is an edge, so is $y \rightarrow x$.

Given any finite, symmetric web graph, let

$$PR(x) ::= \frac{\text{out-degree}(x)}{e},$$

where e is the total number of edges in the graph. Show that this is a solution for the system of equations that are satisfied by the page rank as discussed in section 2 of lecture notes 9.

Problem 6. [25 points] Two students from Podunk University have a neat idea with which they intend to beat out all of the top search engines! Their new product, based on a simple web search algorithm called *Doodle*, uses the following ranking algorithm:

$$\text{Doodlerank}(x) = \sum_{y \rightarrow x} \text{Doodlerank}(y)$$

(the Doodleranks are required to be greater than or equal to 0). This is much nicer than Pagerank, since it gets rid of that silly weighting scheme!

(a) [5 pts] Describe the set of possible settings of Doodlerank's for the nodes in the following graphs.

1. The directed path of length n .
2. The directed cycle of length n .

(b) [5 pts] Give an example of a graph in which each node can reach any other node, but for which the only way to assign weights so that the Doodlerank equations are satisfied is so that the Doodlerank weights are all zero!

Ok, the Podunk students are finally convinced that they have to use the weighting scheme from Google – that is, the equations must satisfy

$$\text{Doodlerank}(x) = \sum_{y \rightarrow x} \frac{\text{Doodlerank}(y)}{\text{outdegree}(y)}$$

However, the Podunk students want to make their fortune by skipping any modification of the original graph. Unlike Pagerank, the sinks in Doodlerank are not made to point to any universal supernodes and maintain an outdegree of 0.

This scheme has also a major problem!

(c) [5 pts] First show by induction that if any node x is assigned Doodlerank 0, then any node y that can reach x in a directed walk must also be assigned Doodlerank 0.

(d) [10 pts] In the next set of problem parts we are going to show that any sink must be assigned Doodlerank 0. To do this, we are going to think of the Doodlerank equations as describing votes by nodes for each of their neighbors. We represent a vote by x for y as a weighted edge $x \rightarrow y$.

1. Suppose V is the set of all the nodes in the graph. Let S be the set of sinks and $T = V - S$ the set of non-sinks. First show that the Doodlerank of a single node $x \in T$ (i.e. $\text{outdegree}(x) \geq 1$) must equal the sum of the weights on the outgoing edges of x .
2. Next show the same property holds over the entire set of non-sinks T by writing the sum of Doodleranks of nodes in T , $\sum_{x \in T} \text{Doodlerank}(x)$, as the sum of the weights of outgoing edges from all nodes $x \in T$ to nodes $y \in V$.
3. Finally, show that any sink must be assigned Doodlerank 0.
Hint: Transform the sum of Doodleranks of non-sink nodes in T , $\sum_{x \in T} \text{Doodlerank}(x)$, into a sum of Doodleranks of all nodes in V , $\sum_{y \in V} \text{Doodlerank}(y)$. Use this to show that the sum of Doodleranks of all sink nodes $z \in S$ must be 0.
4. Finally, conclude that any node which can reach a sink must also be assigned a Doodlerank of 0. So it's not too likely that Podunk U. is going to be hitting up these students for contributions anytime soon!