Advanced Algorithms — Exercise Set 6

- Submit on Gradescope by 1:15pm on **April 7th**.
- Feel free to discuss with others, but write up your own work.
- Half points on this exercise set are awarded for completion / effort. Use it to learn!

Maximal Matchings

Recall that a matching in an undirected graph $G = (V, E)$ is a set of edges $M \subseteq E$ such that no two edges in M share an endpoint. We saw earlier in the course that finding the maxmimum cardinality matching in a graph is solvable in polynomial time.

The thing is, approximation algorithms are useful even for problems in **P**. This is mostly because they are faster and simpler, but also because they often perform better in practice than their worst-case guarantees might lead us to believe.

Here, we will see a simple 1/2-approximation for maximum matching in an arbitrary (not necessarily bipartite) graph. Consider the following algorithm.

> Start with $M = \emptyset$.
>
> While there are edges left:
>
> 1. Pick any remaining edge $e = \{u, v\}$,
> 2. Add e to M,
> 3. Delete e from the graph and all edges that touch u or v.
>
> Return M.

Problem 1. Run this algorithm on the graph in Figure 1. How many edges are in the matching that is returned? How many edges are in the maximum matching?

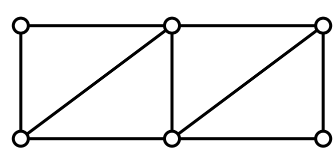

Figure 1: A graph.

Problem 2. A matching is **maximal** if no edges can be added to it while preserving the fact that it's a valid matching. See Figure 2 for some examples. In some sense, a maximal matching is a **local optimum** when searching for a maximum matching. Explain why the algorithm above always produces a maximal matching.

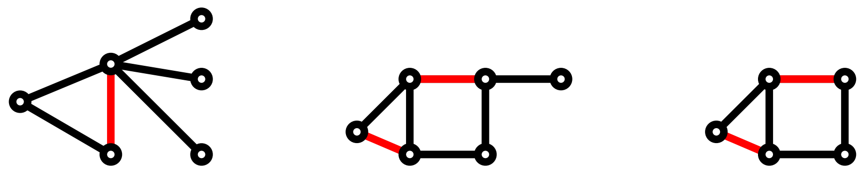

Figure 2: Maximal Matchings

Problem 3. Show the following fact: in any graph, the number of edges in any maximal matching is at least half of the number of edges in the maximum matching.

[Hint: Consider a maximal matching M. Show that there can be at most two edges in the optimal matching for every edge in M.]

Problem 4. Conclude that this algorithm is a 1/2-approximation for max. cardinality matching.

Problem 5 (Bonus)**.** Extend the above algorithm to the setting where there are weights on edges and the goal is to find a maximum weight matching. Explain why your algorithm is yields a 1/2-approximation for the weighted problem as well.

Final Project Presentations

Problem 6. Read the description of the final projects posted on the couse website. Start thinking about what you would like to do! Write here at least two tentative ideas, who you might want to work with if anyone, and at least one question you have about the project. Give as much detail as you can at this stage.