Advanced Topics in Algorithms
Computer Science 181b
Spring 2012
Homework 2, Due Wednesday, February 1

1. **[15 Points] That Weird Inequality!** Consider an arbitrary graph $G = (V, E)$ and a matching M in that graph. Show that for any $U \subseteq V$

$$|M| \leq \frac{1}{2}(|V| + |U| - o(G - U))$$

where $G - U$ is the graph induced by removing the vertices in U from G and $o(G - U)$ denotes the number of connected components in $G - U$ that have an odd number of vertices.

2. **[15 Points] Edmond's Algorithm, Part 1.** Let G be an undirected graph and let M be some matching in G. Let p be an even length alternating path beginning with an unmatched vertex u and ending at a matched vertex v (this is called the "stem"), where v is a vertex on an odd length cycle of length $2k + 1$ that contains k matched edges (the cycle is called the "blossom"). The matching M has the same size as a matching M' in which the edges on the stem are switched so that unmatched-matched-unmatched-matched, etc. is replaced by matched-unmatched-matched-unmatched. In this way, the first vertex u on the stem becomes matched and stem edge ending at v becomes unmatched, which just makes your analysis a tad bit easier. (Notice that M' is a maximum matching iff M is a maximum matching, so M' is just as good as M.) Now, we wish to show that M' is a maximum matching in G iff $M' - B$ (the matching M' but with blossom edges removed) is a maximum matching in $G - B$ (the graph G with the blossom replaced by a single vertex). This is a short and sweet proof and you'll need this result in the next part.

3. **[25 Points] Edmond's Algorithm, Part Deux!** Recall that Edmond's Algorithm works like this: Starting with a graph G and some matching (starting with the empty matching is fine), label all of the unmatched vertices EVEN and clear the labels from all other vertices. Then, repeatedly choose some some EVEN vertex u to process.

   - If there is an edge (u, v) where v is unlabelled, label v ODD. Since v must be a matched vertex, follow the matched edge incident on v to its matched neighbor w. Label w EVEN. We have now extended our augmenting path "forest".

- If there is an edge (u, v) where v is labelled EVEN and v was reached from a different root, then we have found an augmenting path. Use that augmenting path to make the matching larger. Then, expand any blossoms that got shrunk in order to bring this new matching back to the original graph G. Finally, start this entire process from scratch using the new matching in graph G.
- If there is an edge (u, v) where v is labelled EVEN and v was reached from the same root as u then we have found a blossom. Flip the stem (as described in the previous problem), contract the blossom, and start over in this new smaller graph.

(a) Show that the labeling process is consistent in the sense that a vertex that is currently labelled EVEN cannot later be labelled ODD by another vertex or vice versa.

(b) At each step, the algorithm either increases the size of the augmenting path forest, finds an augmenting path and starts all over with the larger matching, or shrinks a blossom and starts the search anew in the contracted graph. If none of these cases can be applied, we **claim** that our matching is a maximum matching in the current graph (which is the original graph but with some blossoms possibly shrunk). Prove this claim. To do so, you'll need to appeal to the result of Problem 1 and choose U to be the set of ODD labelled vertices.

(c) Explain why this matching, with the blossoms expanded and "every other edge" in each blossom added to the matching as we expand, results in a maximum matching for the original graph G.

(d) Derive a polynomial-time upper bound on the worst-case running time of Edmond's Algorithm.

4. **[20 Points] The Tutte-Berge Formula and Tutte's Matching Theorem!** The Tutte-Berge Formula is yet another min-max formula (like min cut-max flow, min vertex cover-max matching) and it states:

$$max_M|M| = \min_{U \subseteq V} \frac{1}{2}(|V| + |U| - o(G - U))$$

(a) Prove this inequality by using the correctness of Edmond's Algorithm from the previous problem.

(b) Tutte's Theorem states that a graph G has a perfect matching (a matching of all vertices) iff for every set U, the number of odd components of $G-U$ is at most $|U|$. Prove it.

5. **[25 Points] The Stable Marriage Problem.** The *stable marriage problem* is as follows:[1] We are given a completely connected bipartite graph with n suitors and n hosts. Each suitor has its own personal total ranking of the hosts and each host has its own total ranking of the suitors. (That is, each suitor ranks the n hosts with distinct numbers between 1 and n and the same is true for hosts ranking suitors.) The objective is to find a perfect matching of the suitors and hosts that has no *instability*: An instability arises if suitor s is not matched to host h but s prefers h to its mate and h prefers s to its mate. A perfect matching with no instability is called a *stable marriage*.

   The Gale-Shapley stable marriage algorithm works as follows: Initially, all suitors and hosts are marked as unengaged. At each round, we let the unengaged suitors take turns making engagement offers. An unengaged suitor makes an offer to the highest rank host that has not previously rejected it. If the host that receives the offer is not yet engaged, it accepts the engagement offer. If the host is already engaged, then it accepts the engagement if the host prefers the new suitor to the suitor to whom it is already engaged. In that case, the previous suitor is rejected and becomes unengaged. Otherwise the host rejects the offer and the suitor remains unengaged in this round. (Notice that a suitor can be rejected by a host immediately or it can be temporarily engaged to a suitor and rejected later for a better suitor.) The algorithm terminates if all parties are engaged or when all suitors can make no more engagement offers. When the algorithm terminates, all of the engaged couples get married.

   (a) Prove that this algorithm terminates and give the worst case number of rounds until termination occurs.

   (b) Prove that upon termination, everyone is married and this is a stable marriage.

   (c) In general, the stable marriage constructed by this algorithm is not necessarily the same as the one that is found if the roles of the hosts and suitors are reversed. Construct a small case in which the stable marriage that results from this algorithm is different from the stable marriage that

[1] This is the gender-neutral description of the problem. In the traditional description, the men are the suitors and the women are the hosts.

results if the roles are reversed and the hosts make the offers to the suitors. This also implies that a stable marriage is not unique.

(d) The Gale-Shapley algorithm has some inherent nondeterminism because we did not specify the order in which the suitors make offers in each round. It is possible that the order in which offers are made can result in different stable marriages. In fact though, the algorithm is *robust* in the sense that it finds the same solution regardless of the order in which the suitors take turns making offers. In addition, the algorithm finds a *suitor-optimal* stable marriage in the sense that it finds a matching M such that for any other stable matching M' and any suitor s, s either gets married to the same host in M and M' or s gets a host in M that it prefers to the host it gets in M'. Amazing!

  i. Explain why showing that the Gale-Shapley algorithm finds a suitor-optimal stable marriage implies that the algorithm is robust.

  ii. Prove that the Gale-Shapley algorithm always finds a suitor-optimal stable marriage. (For terminological convenience, you may wish to say that suitor s and host h are *stable partners* if there exists some stable marriage in which s and h are matched.)