CS420/520: Graph Theory with Applications to CS, Winter 2020

Homework 3

Due: Tue, 2/20/20

Homework Policy:

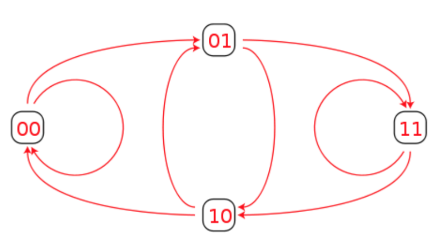
1. Students should work on homework assignments in groups of preferably three people. Each group submits to TEACH one set of typeset solutions, and hands in a printed hard copy in class or slides the hard copy under my door before the midnight of the due day. The hard copy will be graded.
2. The goal of the homework assignments is for you to learn solving algorithmic problems. So, I recommend spending sufficient time thinking about problems individually before discussing them with your friends.
3. You are allowed to discuss the problems with other groups, and you are allowed to use other resources, but you *must* cite them. Also, you must write everything in your own words, copying verbatim is plagiarism.
4. *I don't know policy:* you may write "I don't know" *and nothing else* to answer a question and receive 25 percent of the total points for that problem whereas a completely wrong answer will receive zero.
5. Algorithms should be explained in plain english. Of course, you can use pseudocodes if it helps your explanation, but the grader will not try to understand a complicated pseudocode.
6. **Solutions must be typeset.**

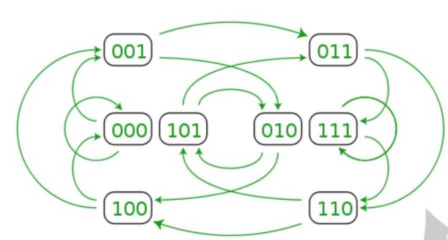
Problem 1.

(a) Let u and v be two non-adjacent vertices of a graph $G = (V, E)$, such that $deg(u) + deg(v) \geq V$. Show that G is Hamiltonian if and only of $G + uv$ is Hamiltonian.

(b) Show that any graph with n vertices and minimum degree $\geq n/2$ is Hamiltonian.

Problem 2. De Bruijn's graph on binary strings of length ℓ is defined as follows. It has 2^ℓ vertices, one for each binary string of length ℓ. There is a directed edge $u \rightarrow v$ if and only if the last $\ell - 1$ bits of u and the first $\ell - 1$ bits of v are identical. Here are De Bruijn graphs for $\ell = 2$ and $\ell = 3$.





(a) Show that the De Bruijn graph is Eulerian.

(b) Show that the DeBruijn graph is Hamiltonian.

(c) Given a positive integer m find the smallest binary string that contains all binary strings of length m as substrings. For example 01100 contains all binary strings of length 2, and 0001011100 contains all binary strings of length 3.

Problem 3. Describe and analyze an algorithm to find the second smallest spanning tree of a given graph G, that is, the spanning tree of G with smallest total weight except for the minimum spanning tree. You can assume that the edge weights are distinct.

Problem 4.

(a) A matching is maximal if it does not leave any free edge. Let M be a maximal matching and M^* be a maximum matching. Prove that $|M| \geq \frac{1}{2} \cdot |M^*|$. (Note that this implies a linear time 1/2-approximation algorithm for computing the maximum matching. What is that algorithm?)

(b) Suppose the degree of all vertices is smaller than a constant Δ. Design an $O(V)$ time algorithm to find a matching M' such that $|M'| \geq \frac{2}{3}|M^*|$. What is the running time of your algorithm as a function of Δ and V?

Problem 5. Let $G = (V, E)$ be a graph and let M be a matching in G. Suppose that v is unmatched by M and that there is no augmenting path with respect to M that starts at v. Show that there exists a maximum matching M^*. in which v is unmatched.