CSC 2420 Spring 2016, Assignment 2
Due date: TBA

NOTE: If you are taking the course for credit, then you may either work by yourself or with at most one other student taking the course for credit. You must specify with whom you are collaborating and the extent of collaboration. It is certainly preferable for you to solve the questions without consulting a published source. However, if you are using a published source then you must specify the source and you should at least try to improve upon the presentation of the result.

1. Let $G = (V, E)$ be a node weighted graph where $w(v)$ denotes the weight of $v \in V$, $w(S)$ denotes $\sum_{v_i \in S} w_i$ and $Nbhd(v) = \{v' | (v, v') \in E\}$ is the set of adjacent vertices (the neighbourhood) of v. Consider the following local search algorithm for the weighted max independent set (WMIS) problem.

   $S := \emptyset$
   While $\exists v \in V$ such that $w(S + \{v\} - Nbdh(v)) > w(S)$
   $\quad S := S + \{v\} - Nbhd(v)$
   End While

   Show that this local search algorithm is a k-approximation algorithm when G is a $k+1$-claw free graph. Show that the bound is tight.

   Do not worry about the convergence time. Essentially show that the locality gap is at most k.

2. Suppose we are given an n by n grid where each cell contains a letter. We want to form a sequence of adjacent cells forming the word "ALGORITHM". Cells are considered adjacent if they share an edge.

   - Give an algorithm that will determine in time polynomial (in n) whether or not there is such a sequence. Briefly explain why your algorithm will provide the correct answer.
   - We would like to form as many cell disjoint sequences as possible forming the word ALGORITHM. Give an algorithm to solve this problem in time polynomial in n. Briefly explain why your algorithm will provide the correct answer.

3. Consider the following call routing makespan problem. There is an n node bi-directional ring network $G = (V, E)$ upon which calls $\{C_1, C_2, \ldots, C_t\}$ must be routed. That is, $V = \{0, 1, \ldots, n-1\}$ and $E = \{(i, i+1 \ mod\ n)\} \cup \{(i, i-1 \ mod\ n)\}$ and calls C_j are pairs (s_j, f_j) originating at node s_j and terminating at node f_j. Each call C_j can be routed in a clockwise or counter-clockwise direction and incurs a routing cost p_j in the clockwise direction and cost q_j in the counter-clockwise on each

directed edge it uses. The load L_e on any *directed edge* e is the sum of the routing costs for all calls using that edge. The goal is to minimize $\max_e L_e$.

- Formulate this problem as an IP. Indicate the intended meaning of each variable in the IP.
- Using an LP relaxation of this problem, show how to derive a constant approximation algorithm. What is the constant you obtain?

4. Consider the maximum matching problem. That is, given a graph $G = (V, E)$, find a subset of edges $E' \subseteq E$ such that for all nodes $u \in V$, the degree of u in $G' = (V, E')$ is at most 1. Let $IN(u) = \{e : e = (u, v) \in E \text{ for some } v \in V\}$. We can express the maximum matching problem as the following natural IP:
maximize $\sum_{e \in E} x_e$
subject to : $\sum_{e:e \in IN(u)} x_e \leq 1$ for all $u \in V$
$x_e \in \{0, 1\}$

   Consider the LP relaxation $\mathcal{P}$ (in standard form) of this IP; that is, :
maximize $\sum_{e \in E} x_e$
subject to : $\sum_{e:e \in IN(u)} x_e \leq 1$ for all $u \in V$
$x_e \leq 1$
$x_e \geq 0$

   - State the dual $\mathcal{D}$ of the primal $\mathcal{P}$ using dual variables y_u for $u \in V$.
   - Can you explain this dual as the relaxation of a known optimization problem?

5. Consider a class of graphs $\mathcal{G}$ which is closed under removal of edges and nodes. For example, the following are such classes: acyclic graphs, graphs that can be coloured with k colours, graphs with maximum degree d. Suppose there was a polynomial time algorithm $\mathcal{A}$ for computing the size of a minimum vertex cover for such a class of graphs $\mathcal{G}$. Show how you can use algorithm $\mathcal{A}$ as a subroutine so as to be able to compute an optimal vertex cover for the class $\mathcal{G}$.

6. Given what you have learned so far in this course and this assignment, show how you would derive an optimal algorithm for the vertex cover problem for bipartite graphs.

7. Bonus question. (This is a bonus question for which we do know the answer.)

   Let G be a bipartite graph, with one side labelled side 1 and the other side labelled side 2. Consider the following game between two players: player 1 begins by choosing a vertex u in side 1 and crossing it out. Then player 2 must choose an adjacent (to u) vertex v (in side 2) and crosses it out. And so on: at any point in time, the player must choose a vertex from his side that has not yet been crossed out and that is adjacent to the last vertex picked by the other player. The first player to run out of moves loses. Give a polynomial time algorithm which for any input bipartite graph G will determine which player has a winning strategy. Justify the correctness of your algorithm.