Solutions to Homework 5

CSE 373, Prof. Skiena

May 8, 2006

- **1.**

  (x+y+v1)(z+v2+ v1)(w+v3+ v2)(u+ v+v3)(x+ y+v4)(z+v5+ v4)(w+v6+ v5)
  (u+v+ v6)(x+ y+v7)(z+v8+ v7)(w+v9+ v8)(w+v9+ v8)(u+ v+ v9) (x+y+v10)(x+y+ v10)

- **2.** Can't draw, but it has 4 clause gadgets each with 3 vertices, three variable gadgets each with 2 vertices, and a total of 3*1 + 4*(3+3) edges.

- **6-1.**The general vertex cover problem is known to be NP complete. The graph obtained by reduction from the 3-SAT problem may be converted into an even degree graph by adding a new edge between any two vertices if the original graph contained an edge between these two vertices. Now if a vertex cover is found in this multigraph in polynomial time then a vertex cover in the general graph may also be found in polynomial time, which is impossible. Hence proved.

- **6-2.**The vertex cover problem is known to be NP complete. So we'll aim to get a reduction of all instances of the vertex cover problem to an instance of the setcover problem. Given a graph $G(V, E)$, label all the edges with the elements of the universal set X. For each vertex contruct a set comprising the elements labelling the edges incident upon it. Finding a setcover in polynomial time would result in solution of the vertex cover problem in polynomial time, because the vertices corresponding to the subsets in the cover form the vertex cover. If any edge e is not covered that means that the element labeling the edge is also not covered by the setcover and vice-versa.

- **6-3.**This problem is similar to the previous problem except that the players represent the edges and packets are the vertices. For any vertex cover problem the previous reduction may be used to create an instance of the given problem.

- **6-4** Any hamiltonian circuit problem in a 3 degree regular graph may be reduced to an instance of a eularian cycle problem. Note that the necessary and sufficient condition for a graph to have an eularian cycle is that all vertices should be of even degree. In a 3 degree regular graph, the maximal eularian subgraph covers all the vertices because the number of vertices is even and removing an edge from pairs of vertices gives an even degree graph. Also since all vertices are of degree 3, the eularian cycle can enter and leave a vertex just once. Hence the solution to the maximal eularian sub-graph is the same as the hamiltonian cycle.

- **6-5.**

  (a)The low-degree spanning tree procedure may be called with $k = 2$ to solve the hamiltonian path problem and we know that the latter is not solvable in polynomial time.

  (b) Find the highest degree vertex in the graph. From the adjacency list representation this is possible in $O(E)$ time. If the degree is at least k then there is a such a spanning tree. The spanning tree is same as the BFS tree starting at the vertex of degree greater than k.

- **5.**We show that hamiltonian cycle is reduced to TSP. Let G=(V,E) be an instance of hamiltonian cycle. We construct an instance of TSP as follows. We form the complete graph G'=(V,E'), where E'= (i,j) — i,j E V and i!=j, and we define the cost function c by

$$c(i,j) = \begin{cases} \text{0, if (i,j) is an element of E,} \\ \\ \text{1, otherwise} \end{cases}$$

  The instance of TSP is then (G', c, 0), which is easily formed in polynomial time. Let's show graph G has a hamiltonian cycle iif graph G' has tour of cost at most 0. Assume graph G has hamiltonian cycle h. Each edge in h belongs to E and has cost 0 in G'. Thus, h is a tour in G' with cost 0. Conversly, suppose that graph G' has a tour h' of cost at most 0. Since the costs of the edges in E' are 0 and 1, the cost of tour h' is exactly 0 and each edge on the tour must have cost 0. Therefore, h' contains only edge in E. We conclude that h' is a hamiltonian cycle in G.

- **6.**We need to show that we can increase the degree of low degree vertices so that we do not create any new large cliques. If we seek a k-regular graph, we must create a new k-regular subgraph gadget that does not contain a large clique; One approach would be to try creating a *D-regular* graph where D is the sum of the degrees di of all nodes. For all vertices add $(D - di)$ new *red* nodes and connect them to the original vertex. In all we added $(D-d1)+(D-d2)....+(D-dn) = nD-(d1+d2+....+dn) = nD-D = (n-1)D$ *red* nodes each with degree 1. Thus all the original nodes in the graph have degree D. Now we want to increase the degrees of *red* nodes from 1 to D. We do this by forming $(n - 1)$ groups of size D. For each group we add $(D - 1)$ new *blue* nodes and form a complete bipartite graph with the D *red* nodes in the group. Each of the $(D - 1)$ *blue* nodes is connected to D *red* nodes hence degree D, and each of the D *red* nodes with degree 1 is connected to $(D-1)$ *blue* nodes to make their degree also D. Thus we were succesful in making the whole graph *D-regular*. Claim : We did not add any cliques of size greater than 2 i.e. an edge. This is because we avoided adding any triangles by using the bipartite graph between the *red* and *blue* nodes. Hence a clique in the new graph will be of the same size as the original graph and since finding clique in the original graph is NP-complete we show that the problem of finding clique in k-regular graph is also NP-complete.

- **7.**Any vertex of degree lower than k cannot be in the subgraph, so delete them. If this causes another vertex to be of low degree, delete it, and repeat until no such vertex exists. This is the maximum induced subgraph. To do this efficiently, we represent the graph as an adjacency list where there is a pointer between both copies of each edge (i,j) and (j,i), with a field for each vertex recording its degree. When we delete an edge, we lower the vertex degree, and when it gets below k we put that vertex on a queue or stack for deletion. The time will be linear in the number of edges/vertice in the graph, because each vertex deletion takes time proportional to its degree and each edge is only deleted once.