CSC373— Algorithm Design, Analysis, and Complexity — Spring 2018

Tutorial Exercise 10: Cliques and Intersection Graphs for Intervals

---

1. **Clique.** Given an undirected graph $G = (V, E)$ a clique (pronounced "cleek" in Canadian, heh?) is a subset of vertices $K \subseteq V$ such that, for every pair of distinct vertices $u, v \in K$, the edge (u, v) is in E. See Clique, Graph Theory, Wikipedia.

   A second concept that will be useful is the notion of a complement graph G_c. We say G_c is the complement (graph) of $G = (V, E)$ iff $G_c = (V, E_c)$, where $E_c = \{(u, v) \mid u, v \in V, u \neq v, \text{ and } (u, v) \notin E\}$ (see Complement Graph, Wikipedia). That is, the complement graph G_c is the graph over the same set of vertices, but it contains all (and only) the edges that are not in E.

   Finally, a clique $K \subset V$ of $G = (V, E)$ is said to be a **maximal clique** iff there is no superset W such that $K \subset W \subseteq V$ and W is a clique of G.

   Consider the decision problem:

   Clique: Given an undirected graph and an integer k, does there exist a clique K of G with $|K| \geq k$?

   Clearly, Clique is in NP. We wish to show that Clique is NP-complete. To do this, it is convenient to first note that the independent set problem IndepSet(G, k) is very closely related to the Clique problem for the complement graph, i.e., Clique(G_c, k). Use this strategy to show:

$$\text{IndepSet} \;\equiv_p\; \text{Clique}. \tag{1}$$

2. Consider the interval scheduling problem we started this course with (which is also revisited in Assignment 3, Question 1). That is, suppose we are given a set of jobs $J(k)$ which start at time s_k and end at time f_k, where $s_k < f_k$ are non-negative integers for $k = 1, 2, \ldots, K$. We assume all the finish times are distinct. We wish to schedule a maximum-size subset of these jobs such that no two scheduled jobs overlap in time (i.e., the open time intervals (s_j, f_j) and (s_k, f_k) do not intersect for any pair of jobs $J(j)$ and $J(k)$ that are scheduled).

   Consider the intersection graph for this problem, say $G = (V, E)$, where the vertices, say v_n, are in one to one correspondence with the jobs, $J(n)$, and there is an edge $(v_n, v_m) \in E$ iff $n \neq m$ and jobs $J(n)$ and $J(m)$ intersect (i.e., $(s_n, f_n) \cap (s_m, f_m) \neq \emptyset$). In A3Q1 we show:

$$\text{IntervalSched}(\{J(n)\}_{n=1}^{N}) \leq_p \text{searchIndepSet}(G).$$

   Specifically, we show that finding a solution to the interval scheduling problem is equivalent to finding a maximum-sized independent set for G. Something must be very special about these graphs G to allow for the poly-time greedy solution. We reconsider this issue, this time with the concept of cliques in hand.

   (a) Choose any example of jobs $\{J(k)\}_{k=1}^{K}$ for the interval scheduling problem. Draw the corresponding intersection graph $G = (V, E)$. Find the set of all maximal cliques of G. (To find one maximal clique, you can pick any vertex v and set $T = \{v\}$. Pick any other vertex u such that there are edges from u to all the vertices in T. Add such a u to T, and repeat. When there is no such u, T is a maximal clique. To get the set of all maximal cliques, you need to do this using all possible ways of picking vertices to add to T.)

   (b) Let $\mathcal{K} = \{K_1, K_2, \ldots, K_m\}$ be the set of the maximal cliques found in part (a). Draw the intersection graph $H = (\mathcal{K}, E_K)$ where each vertex of H is a maximal clique of G (i.e., an element of $\mathcal{K}$), and $(K_i, K_j) \in E_K$ iff $K_i \cap K_j \neq \emptyset$ and $K_i \neq K_j$. Then we have the following claim:

   Claim 1: The graph H is a forest.

   Is this claim true for your example?

   (c) Suppose $G = (V, E)$ is a graph such that H, the intersection graph of the maximal cliques of G, satisfies Claim 1 above. Devise a polynomial time greedy algorithm to find a maximum-sized independent set for G. Do you see why it is correct? (You do not need to prove it.)