Solutions to End-Semester Exam Problems

Q1(a) Consider the standard perfect elimination ordering of an interval graph obtained by ordering the intervals in non-decreasing order of finish point. Then if $i < j < k < l$ are intervals in this order, and i is adjacent to k and l, then the start points of k and l must be less than or equal to the finish point of i. Then the finish point of j must be contained in both k and l. Thus j is adjacent to both k and l. Note that this is a stronger property than what is required for strongly chordal graphs.

Q1(b) Any tree is a strongly chordal graph because there do not exist any $i < k < l$ in a perfect elimination ordering such that v_i is adjacent to v_k and v_l. Thus any tree that is not an interval graph will do. The simplest example is obtained by subdividing each edge of $K_{1,3}$ exactly once.

Q1(c) It can be seen that in a strongly chordal graph, if the neighbors of v_1 are $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$, with $i_1 < i_2 < \cdots < i_k$, then $N(v_{i_1}) \subseteq N(v_{i_2}) \subseteq \cdots \subseteq N(v_{i_k})$. Thus any chordal graph that does not have such a simplicial vertex is an example. The simplest one is the graph with 6 vertices in which the maximal cliques are $\{a, b, c\}, \{a, b, d\}, \{a, c, e\}, \{b, c, f\}$.

Q1(d) The same property as mentioned above enables finding a minimum dominating set in a strongly chordal graph. The greedy algorithm is to select a vertex in the dominating set only when you have to. For $i = 1$ to n, select v_i if there exists a vertex $v_j, j \leq i$ such that v_j is not dominated by already selected vertices, and v_j cannot be dominated by any vertex greater than i, that is i is the largest numbered vertex in $N(v_j) \cup \{v_j\}$. To prove the correctness, suppose there exists an optimal dominating set that agrees with the greedy solution up to $i - 1$.

Suppose v_i is included in greedy but not in optimal. Then there exists a vertex v_j with $j \leq i$ that is not dominated by selected vertices less than i, and for which v_i is the largest vertex in $N(v_j) \cup \{v_j\}$. But this implies v_j is not dominated by any vertex in the optimal, a contradiction.

Suppose v_i is not included in greedy but is included in optimal. Remove v_i from the optimal, which will create a set S of undominated vertices. We show that we can get a new dominating set by replacing v_i by a vertex greater than i. Let j be the smallest vertex in S. If j is greater than i, then every vertex in S must be adjacent to v_j, by the perfect elimination ordering. Thus replacing v_i by v_j gives a new dominating set that agrees with greedy up to i. If $j \leq i$, since j is not dominated by vertices less than i in optimal, and hence also in greedy, and v_i is not selected in greedy, v_j must have a neighbor v_k with $k > i$. We claim that every vertex in S is adjacent to v_k. By the perfect elimination ordering, v_i is adjacent to v_k. So any vertex in S that is greater than i is adjacent to v_k. On the other hand any vertex v_l in S other than v_j that is less than i, must satisfy $j < l < i < k$. Now, $v_j v_i, v_j v_k, v_l v_i$ are edges, and by the strongly chordal property $v_l v_k$ is also an edge. Thus all vertices in S are adjacent to v_k, and replacing v_i by v_k we get a new optimal dominating set that agrees with greedy up to i.

Q2. Suppose the vertices are numbered in the perfect elimination ordering. Then every vertex has at most two neighbors with a larger number. For every edge, we maintain a weight that indicates the length of the longest path between its endpoints, whose internal vertices have been deleted. We also keep track of the largest cycle found so far. Initially, each edge has weight 1. Consider vertices in the perfect elimination ordering $v_1, v_2, \ldots, v_n$. At stage i, we delete vertex v_i and update weights in the remaining graph. If v_i has degree at most 1, simply delete it. Suppose v_i has two neighbors v_j, v_k such that $i < j < k$. Then we get a cycle of length $w_{ij} + w_{ik} + w_{jk}$. Update the current longest cycle if needed. The new weight of the edge w_{jk} is $\max(w_{jk}, w_{ij} + w_{ik})$. This can be done in $O(n)$ time by keeping for every vertex at most two entries, which are the weights of the edges joining it to larger numbered vertices.

Q3(a) Suppose the interval graph is represented by the consecutive ordering of maximal cliques. Let $C_1, C_2, \ldots, C_m$ be the maximal cliques. The optimal solution consists of a collection of disjoint clique components. We consider all possibilities for the leftmost clique in this collection. All vertices in this clique must come from some maximal clique C_i. Suppose this includes a vertex that occurs in C_i and finishes in C_j for some $j \geq i$. Then any vertex contained in C_i that finishes between C_i and C_j can be added to the solution, keeping it P_3-free. Therefore the leftmost clique can be identified by the set of vertices in C_i that finish between C_i and C_j. Here we consider all possibilities for C_i and C_j. This fixes the first clique. Now to find the rest of the cliques, delete all vertices that are adjacent to any vertex in the first clique, that is those that start at any clique less than or equal to C_j. Therefore the problem now reduces to finding the optimal solution for vertices that start later than C_j.

Let $S[i]$ denote the maximum order of a P_3-free induced subgraph of the interval graph induced by vertices that start after C_i. Then we get the recurrence

$$S[i] = \max_{i<j\leq k\leq m} N[i,j,k] + S[k].$$

Here $N[i,j,k]$ is the number of vertices that start after C_i, are contained in C_j and finish between C_j and C_k (inclusive). The maximum is taken over all choices of j,k such that $i < j \leq k$. Initially, $S[m] = 0$. $S[0]$ is the final answer.

Q3(b) The main idea for permutation graphs is similar. The first component in a possible solution will be some descending subsequence. If a_i is the first element of the subsequence and a_j the last, the remaining vertices in the solution can only be vertices that are greater than a_i and appear after a_j. Therefore, the subproblem to be solved is defined by $S[i,j]$, which is the optimal solution for the subsequence of numbers that are greater than a_i and occur after a_j.

The recurrence is

$$S[i,j] = \max_{j<k\leq l, a_k\geq a_l>a_i} N[k,l] + S[k,l]$$

where $N[k,l]$ is the length of the longest descending subsequence that starts with a_k and ends with a_l. The initial condition is $S[n,j] = 0$ for all j. The final answer is $S[0,0]$, where we assume $a_0 = 0$. The $N[k,l]$ values can be computed separately using the standard algorithms.

Q4 Let the optimal numbering of the vertices of G be $v_1, v_2, \ldots, v_n$ and that for H be $u_1, u_2, \ldots, u_m$. Let b_1, b_2 be the bandwidths of G and H obtained by these numberings. Now consider the numberings of $G \vee H$ defined as $f_1 = v_1, v_2, \ldots, v_{\lceil n/2\rceil}, u_1, u_2, \ldots, u_m, v_{\lceil n/2\rceil+1}, \ldots, v_n$ and $f_2 = u_1, u_2, \ldots, u_{\lceil m/2\rceil}, v_1, v_2, \ldots, v_n, u_{\lceil m/2\rceil+1}, \ldots, u_m$. Now for any edge joining a vertex v in G to a vertex u in H, $|f_1(v) - f_1(u)| \leq \lceil n/2\rceil + m - 1$ and $|f_2(v) - f_2(u)| \leq \lceil m/2\rceil + n - 1$. For any edge v_iv_j in G, $|f_1(v_i) - f_1(v_j)| \leq b_1 + m$, and $|f_2(v_i) - f_2(v_j)| \leq b_1$. Similarly, for an edge u_iu_j in H, $|f_1(u_i) - f_1(u_j)| \leq b_2$ and $|f_2(u_i) - f_2(u_j)| \leq b_2 + n$. Thus, considering the labeling f_1, we get $B(G \vee H) \leq \max(b_1, \lceil n/2\rceil - 1) + m$, and by the labeling f_2, $B(G \vee H) \leq \max(b_2, \lceil m/2\rceil - 1) + n$. Hence $B(G \vee H)$ is at most the minimum of these two values.

To prove that this is optimal, consider an optimal labeling f of $G \vee H$. We may assume the first and last vertex are in the same graph, without loss of generality G, otherwise the bandwidth becomes $n + m - 1$. Let u_i be the leftmost vertex of H and u_j the rightmost vertex of H in the ordering.

We show that there exists an edge in $G \vee H$ such that the difference in the numbers of its endpoints is at least $\lceil n/2 \rceil - 1 + m$ and an edge for which the difference is at least $b_1 + m$. If there are at least $\lceil n/2 \rceil$ vertices of G to the left of u_i, then the edge joining the first vertex to u_j has difference at least $\lceil n/2 \rceil - 1 + m$, otherwise the edge joining u_i to the last vertex is the required edge. Since the bandwidth of G is b_1, there exists an edge $v_k v_l$ in G such that there are at least $b_1 - 1$ vertices of G between v_k and v_l, with v_k to the left of v_l. If v_k is to the left of u_i and v_l to the right of u_j, there are at least $b_1 - 1 + m$ vertices between v_k and v_l in the labeling f. If v_k is to the right of u_i, then the edge joining u_i to the last vertex has at least b_1 vertices of G and $m - 1$ vertices of H lying between its endpoints. If v_l is to the left of u_j, then the edge joining the first vertex to u_j is the required edge.

This implies that $B(G \vee H) \geq \max(b_1, \lceil n/2 \rceil - 1) + m$. This proves the formula.