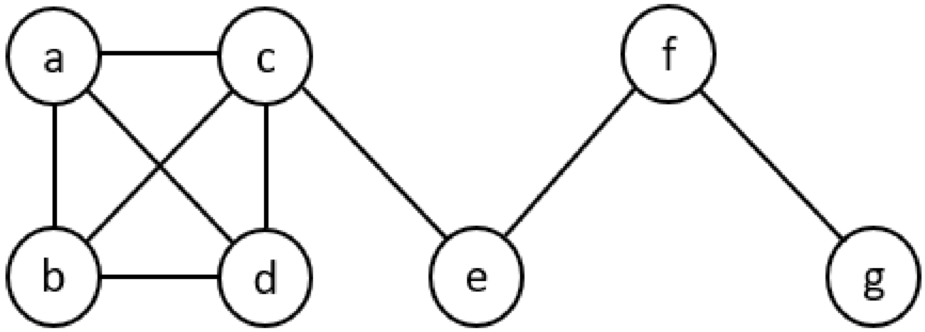
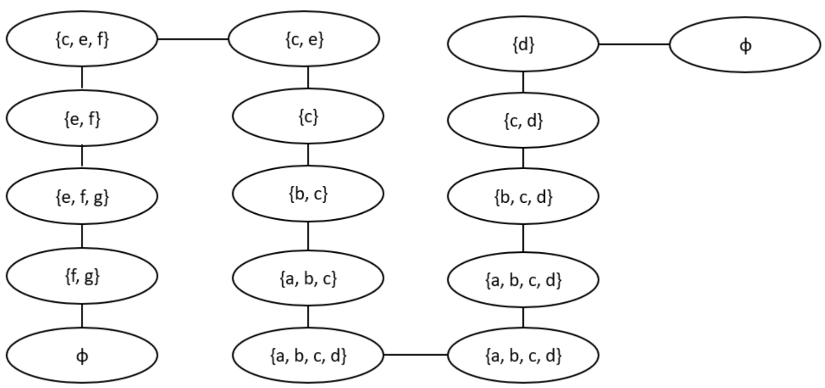


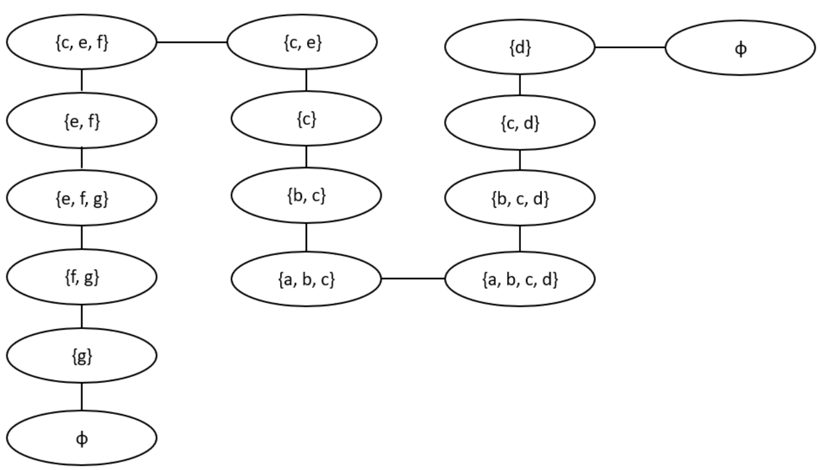
Figure 1: Graph G

Week 6 Assignment

1. (1 point) Consider the graph G shown in figure 1. Which of the following is/are nice tree decomposition(s) of G? In all the figures consider the tree as a path with the rightmost vertex as the root.



A.



B.

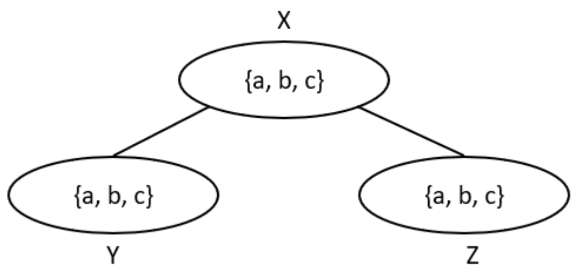
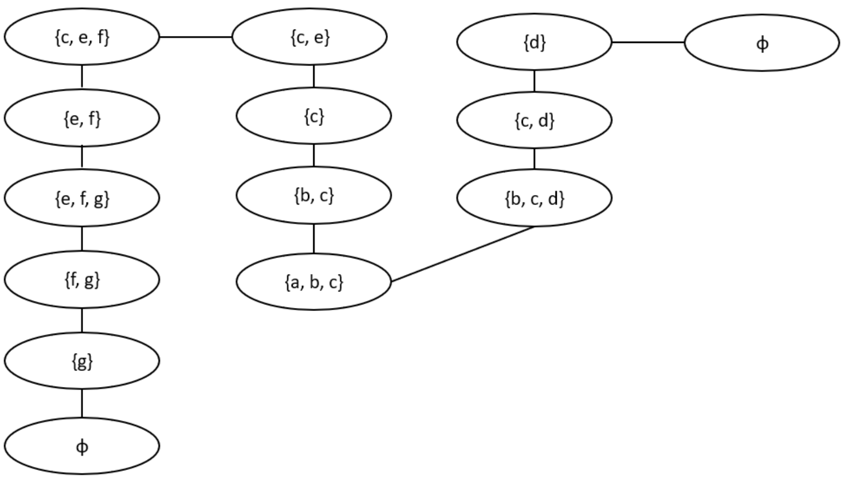


Figure 2: A join node

C. 

Solution:

The option B satisfies all the properties of nice tree decomposition. In option A, the vertices f and g have been simultaneously introduced. In option C, the vertex a has been forgotten in the same bag where vertex d has been introduced. Hence the correct options are A and C.

2. (1 point) Statement I: A vertex can be introduced in exactly one bag in a nice tree decomposition. Statement II: A vertex can be forgotten in exactly one bag in a nice tree decomposition.

   Which of the following is correct?

   A. Both Statement I and II are correct.

   B. Both Statement I and II are wrong,

   C. Statement I is correct but Statement II is wrong.

   D. Statement I is wrong but Statement II is correct.

Solution:

Statement I is wrong because when two same nodes (bags) are joined, the vertices in them have been introduced in some of the descendants of both these bags. So it is possible that a vertex is introduced multiple times. Statement II is correct because once a vertex is forgotten it cannot be introduced again as a tree decomposition needs to satisfy the property that all the bags (nodes) containing a vertex must form a connected subtree.

3. (1 point) In Figure 2 a portion of a nice tree decomposition $\mathcal{T} = (T, \{B_t\}_{\{t \in V(T)\}}$ is shown for some graph G. Here T denotes the vertex set of the tree and B_t denotes the bag containing t. Let V_t denote the union of all bags in the subtree rooted at t including the bag B_t. Consider the dynamic programming algorithm for maximum independent set on graphs of bounded treewidth. The entry $m[t, S]$ denotes the weight of the maximum independent set of the graph $G[V_t]$ whose intersection with B_t is exactly S. If S is not an independent set, $m[t, S] = -\infty$. Also we know that a is adjacent to b and c is non-adjacent to both a and b in G. Let $w(v)$ denote the weight of the vertex v for any $v \in V(G)$. Which of the following statement(s) is/are true?

   A. $c[X, \{a, b\}] = -\infty$**.**

   B. $m[X, \{a, c\}] = \max\{m[Y, \{a, c\}], m[Z, \{a, c\}\}] + w(a) + w(c)$.

   C. $m[X, \{a, c\}] = m[Y, \{a, c\}] + m[Z, \{a, c\}] - w(a) - w(c)$**.**

   D. $m[X, \{a, c\}] = m[Y, \{a, c\}] + m[Z, \{a, c\}] + w(a) + w(c)$.

   E. $m[X, \{a, c\}] = \max\{m[Y, \{a, c\}], m[Z, \{a, c\}\}] - w(a) - w(c)$.

   Solution:

   If S is not an independent set, there cannot be any maximum independent set which contains S therefore $m[t, S] = -\infty$ for any t. Hence option A is correct.

   For a join node X with children Y and Z, the maximum independent set in the graph $G[V_X]$ which intersects the bag B_X at exactly S for a given independent set S is given by the union of the maximum indepedent set in the graph $G[V_Y]$ which intersects the bag B_Y at exactly S and the maximum indepedent set in the graph $G[V_Z]$ which intersects the bag B_Z at exactly Z. Therefore for any independent set S which is a subset of B_X, we have $m[X, S] = m[Y, S] + m[Z, S] - w(S)$ where $w(S)$ is the sum of the weights of all the vertices in S. We subtract $w(S)$ because the vertices in S are counted twice (once in the maximum independent set of $G[V_Y]$ and once in the maximum independent set of $G[V_Z]$). Therefore option C is correct and B, D, E are wrong.

4. (1 point) Consider the maximum weight independent set problem and the same notations in Question 3. But suppose this time we have a forget node $W = \{e, f\}$ with a child $U = \{e, f, g\}$. Suppose the vertices e, f and g are pairwise non-adjacent in G. Which of the following statement(s) is/are correct?

   A. $m[W, \{e\}] = m[U, \phi] + w(e)$.

   B. $m[W, \{e\}] = m[U, \{e\}]$.

   C. $m[W, \{e\}] = \mathbf{max}\{m[U, \{e\}], m[U, \{e, g\}]\}$**.**

   D. $m[W, \{f\}] = \mathbf{max}\{m[U, \{f\}], m[U, \{f, g\}]\}$**.**

   Solution:

   A maximum weight independent set of $G[V_W]$ whose intersection with B_W is $\{e\}$ is an independent set which has the maximum weight among all the independent sets among all the independent sets of $G[V_U]$ whose intersection with B_U is {e} and all the independent sets of $G[V_U]$ whose intersection with B_U is {e,g}. Thus A and B are wrong and C is correct. Option D is also correct by the same argument using f instead of e.

5. (1 point) Is the following statement true or false: For a graph G, consider a separator $S \subset V(G)$ such that after deleting S, the graph $G \setminus S$ has two (non-empty) components X and Y such that there is no edge between a vertex of X and a vertex of Y. Also, the graph G does not become disconnected after deleting any proper subset of S. A minimum dominating set of G can be obtained by taking the union of a minimum dominating set of $X \cup S$ and a minimum dominating set of $Y \cup S$ which contain the same vertices from S.

   A. True

B. False

Solution:

Suppose the graph G is such that X is a clique with 100 vertices, S is an independent set with 100 vertices and Y is a single vertex. Every vertex in X is adjacent to exactly one vertex in S and all the vertices in S are adjacent to the vertex y in Y. Any dominating set of $X \cup S$ is of size at least 100 and taking a vertex from X and the vertex y from Y forms a minimum dominating set of G of size 2. Thus this set is not formed by taking the union of a minimum dominating set of $X \cup S$ and a minimum dominating set of $Y \cup S$ which contain the same vertices from S. Thus the statement is false.

6. (1 point) Which of the following statement(s) is/are true for the dynamic programming algorithm for Dominating Set discussed in Week 6 Lecture 26?

   A. We attempt to "guess" the 3-partiton of each bag by assigning colors black, white and grey to the vertices such that the black vertices are contained in the solution, the white vertices are not contained in the solution but are dominated by the solution constructed so far and the grey vertices may or may not be dominated by the solution constructed so far.

   B. We attempt to "guess" the 3-partiton of each bag by assigning colors black, white and grey to the vertices such that the black vertices are contained in the solution, the white vertices are not contained in the solution but are dominated by the solution constructed so far and the grey vertices are not dominated by the solution constructed so far.

   C. Let V_x denote the vertices in G which belong to the subtree rooted at x and B_x denote the vertices of G contained in the bag x. Let $M[x, S_1, S_2]$ be the size of a smallest dominating set D such that every vertex in $V_x \setminus B_x$ is dominated by D, $D \cap B_x = S_1$ and the vertices in S_2 are not in D but are dominated by the vertices in D. Then if x is a join node with left child y and right child z, then for computing $M[x, S_1, S_2]$ we need to solve $3^{|S_2|}$ subproblems.

   D. Let V_x denote the vertices in G which belong to the subtree rooted at x and B_x denote the vertices of G contained in the bag x. Let $M[x, S_1, S_2]$ be the size of a smallest dominating set D such that every vertex in $V_x \setminus B_x$ is dominated by D, $D \cap B_x = S_1$ and the vertices in S_2 are not in D but are dominated by the vertices in D. Then if x is a join node with left child y and right child z, then for computing $M[x, S_1, S_2]$ we need to solve $3^{|S_1|}$ subproblems.

Solution:

In the dynamic programming algorithm for Dominating Set discussed in Week 6 Lecture 26, we attempt to guess which of the vertices are already in the solution, which of the vertices are not in the solution but dominated by a partial solution and which of the vertices may not yet be dominated by the solution. Thus option A is correct. We do not prevent the vertices which are in S_3 (the ones which may not be dominated by the vertices in the partial solution) from being dominated by the partial solution. Thus option B is wrong.

We guess whether each vertex which is dominated but not in the solution is dominated by the partial solution for the left child but not by the partial solution for the right child or by the partial solution for the right child but not by the partial solution for the left child or by both. Thus we will need to solve $3^{|S_2|}$ subproblems, i.e., option C is correct and D is wrong.

7. (1 point) Is the following statement true or false: "A graph with treewidth k is $k + 1$ colorable."

   A. True

   B. False

Solution:

Consider a nice tree decomposition of a graph with a treewidth of k and consider the set of colors from $[1, k+1]$. Color all the vertices in the root with distinct colors. Now for coloring the vertices in a child, if a vertex occurs in the parent node then give it the same color as the parent node, otherwise give it the smallest color which is not used by any of the vertices in this node. It can be verified that this coloring is well defined (because the subgraph of the tree with bags that contain any particular vertex of the graph is connected). The coloring cannot use more than $k+1$ colors because the size of a bag cannot be more than $k+1$. Thus the statement is true.

8. (1 point) Is the following statement true or false: "Any bag with $k+1$ vertices in the tree decomposition of width k of a graph G forms a balanced separator of G ($k \geq 2$)." ? (A balanced seperator is a vertex set of G such that after deleting it, every connected component has size at most $|V(G)|/2$.

   A. True

   B. False

Solution:

Consider a graph G with n cliques of size $k > 2$ i.e., $C_1, C_2, \ldots, C_n$ and join a vertex u_i of C_i to a vertex v_{i+1} of C_{i+1} for all $i \in [1, n-1]$ such that $u_i \neq v_i$ for any $i \in [2, n-1]$. Now consider the path decomposition of this graph $P_1 P_2 \ldots P_{2n-1}$ such that $P_1 = V(C_1)$, $P_2 = \{u_1, v_2\}$, $P_3 = V(C_2)$, $P_4 = \{u_2, v_3\}$ and so on. The width of this decomposition is $k-1$. The bag P_1 has size k but it is not a balanced separator. Thus the statement is false.

9. (2 points) Consider the dynamic programming algorithm for the 3−coloring problem on graphs of bounded treewidth. Consider a nice tree decomposition $\mathcal{T} = (T, \{B_t\}_{\{t \in V(T)\}})$. Here T denotes the vertex set of the tree and B_t denotes the bag containing t. Let V_t denote the union of all bags in the subtree rooted at t including the bag B_t. The entry $E[x, c] = 1$ if c is a valid coloring of the vertices in B_x and can be extended to a valid coloring of V_x; and $E[x, c] = 0$ otherwise. Which of the following statement(s) is/are true?

   A. In a node containing a single vertex $E[x, c] = 0$.

   B. In a node containing a single vertex $E[x, c] = 1$.

   C. In an introduce node x with child y and the newly introduced vertex v, $E[x, c]$ will be 1 iff c assigns a color to v which is not assigned to any of its neighbours.

   D. In an introduce node x with child y and the newly introduced vertex v, $E[x, c]$ will be 1 iff c assigns a color to v which is not assigned to any of its neighbours and $E[y, c'] = 1$ where c' is the coloring c restricted to the vertices in B_y.

   E. In a forget node x with child y and the deleted vertex v, $E[x, c]$ will be 1 iff $E[y, c'] = 1$ for all the colorings c' such that c' restricted to B_x is c.

   F. In a forget node x with child y and the deleted vertex v, $E[x, c]$ will be 1 iff $E[y, c'] = 1$ for at least one of the colorings c' such that c' restricted to B_x is c.

   G. In a join node x with children y and z, $E[x, c] = E[y, c] \wedge E[z, c]$.

   H. In a join node x with children y and z, $E[x, c] = E[y, c] \vee E[z, c]$.

Solution:

Any coloring of a single vertex is valid, therefore whenever $B_x = V_x$ contains a single vertex $E[x, c] = 1$. Thus A is wrong and B is correct.

For options C and D, for the coloring c to be valid and extendable, its restriction on B_y must also be valid and extendable and the vertex v should not have the same color as one of its neighbours. Thus C is not correct. Also, if c assigns a color to v which is not assigned to any of its neighbours and is valid and extendable when restricted to the rest of the vertices, then c is valid and extendable. Hence D is correct.

For options E and F, c is a valid coloring of B_x and c can be extended to a valid coloring of V_x if there exists a color from one of the 3 colors such that after giving it to v, the coloring is still valid and can be extended to a valid coloring for the rest of the vertices in V_x. However, this does not mean that any extension of c to a coloring of the vertices in B_y is valid and can be extended to a valid coloring for the rest of the vertices in V_x. Thus E is wrong and F is correct.

For options G and H, in a join node x with children y and z, the coloring c of the vertices in B_x is valid and can be extended to a coloring of the vertices in V_x if and only if the coloring c of the vertices in $B_y = B_x$ is valid and can be extended to a coloring of the vertices in V_y and the coloring c of the vertices in $B_z = B_x$ is valid and can be extended to a coloring of the vertices in V_z. Thus G is correct and H is wrong.