Problem Set 4 - Solutions

1. **Solution:**

   (a) Let $(T, (B_t)_{t\in V(T)})$ be a tree decomposition of G. We show that there exists a bag B_z such that $C \subseteq B_z$. For any edge uv of T, define T^u_{uv} as the connected component of $T - uv$ that contains u. Let W_u be the union of all bags in T^u_{uv} and define W_v analogously. Now notice that at least one of W_u and W_v must contain all the vertices of C: for otherwise, there were vertices $x, y \in C$ such that x appears only in W_u and y appears only in W_v, and thus there would be no bag that contains both of x and y – a contradiction.

   If $C \subseteq W_u$, direct the edge uv towards u; otherwise direct it towards v. Since T is a tree, there is a vertex $z \in V(T)$ such that all the edges incident to z are directed towards z. We claim that, $C \subseteq B_z$. Suppose not; let $x \in C$ be a vertex not in B_z. Consider a neighbor u of z such that x appears in the bags of T^u_{zu}. But since the edge zu is directed towards z, there must also be a bag in T^z_{zu} that contains x. But then in any tree decomposition, the part of the tree that contains x is connected, and so x must appear in $B_z \cap B_u$.

   (b) Consider a tree decomposition of G of width k. Convert it into a *small* tree decomposition of the same width, i.e. a tree decomposition such that there are no two bags B_u and B_v with $B_u \subseteq B_v$. This is always possible by contracting edges uv where $B_u \subseteq B_v$. Consider a leaf z of this small tree decomposition. It must contain a vertex x that appears only in B_z. But then x can have at most k neighbors.

2. **Solution:**

   (a) Let X be an edge-set in the brach-decomposition of G. For each subset A of $\partial(X)$, let $M_X[A]$ denote the minimum-size of a subset of the vertices that covers all edges in X that are not incident to vertices in A. Let X_1, X_2 be the children of X in the decomposition tree. When constructing $M_X[A]$, for every choice of a subset A' of $\partial(X_1) \cap \partial(X_2)$, the cost of using A' for covering all the edges in X not incident to vertices in A is the size of A' plus the cost of covering all edges in X_1 not incident to vertices in A or in A' plus a similar term for X_2. The minimum vertex cover is $M_G[\emptyset]$. the table M is constructed bottom up by calling the following procedure at each node X of the decomposition tree:

   VertexCoverTable(X):
   - if X is a leaf
     - $M_X[A] = 0$ for nonempty A, $M_X[\emptyset] = 1$
   - else
     - let X_1, X_2 be the children of X
     - for every subset A of $\partial(X)$
       - $M_X[A] = \min_{A' \subseteq \partial(X_1)\cap\partial(X_2)} |A'| + \sum_{i\in\{1,2\}} M_{x_i}[(A' \cup A) \cap \partial(X_i)]$
   - return $M_X[\cdot]$

   Computing a single entry of the table takes $O(2^b)$ time. There are $2^b n$ table entires, so the total running time is $2^{O(b)} n$.

   (b) Note that G_{ij} might be disconnected and might have connected components with large radius. However, the radius of the radial graph of each connected component is $O(k)$ since any u-to-v path in the radial graph can be decomposed into a u-to-x path of length $O(k)$, where x is a vertex at level ik, a y-to-v path of length $O(k)$, where v is a vertex at level ik, and a x-to-y path of length 3 that goes through the vertex of the radial graph corresponding to the infinite face of the connected component containing u and v in G_{ij}. Thus, by Tamaki's theorem the branchwidth of G_{ij} is $O(k)$, so the solution from item (a) runs in the desired time bound.