Exercise set 4 Algorithms and Complexity 2023 **Due 13 Nov 2023**

You may collaborate and submit answers in groups of at most three. Good solutions are complete and concise. Please mail to T.vanderZanden@maastrichtuniversity.nl.

8. Design an algorithm that, given a graph G together with a tree decomposition $(T, \{X_t : t \in T\})$ of width w, computes the size of a minimum dominating set in G in $O^*(9^w)$ time.

Hint: You may assume the tree decomposition T is given in nice form. For a given node in $t \in T$, let $G[t]$ denote the subgraph of G induced by the vertices in the bag X_t and all vertices in bags occurring in the subtree of T rooted at t.

Create a dynamic programming table, where $T[t, D, E, I]$ stores, for a node $t \in T$ and partition D, E, I of X_t, the minimum size of a vertex subset $S \subseteq V(G[t])$ such that:

- S dominates every vertex of $G[t]$ except those in E;
- $S \cap X_t = I$.

Show how the dynamic programming tables should be initialized if $t \in T$ is a leaf node ($|X_t| = 1$ and t is a leaf of T). Note that $G[t]$ consists of an isolated vertex.

Show how, if $t \in T$ is an introduce node (t has one child $t' \in T$ and there is exactly one $v \in V$ for which $X_t = X_{t'} \cup \{v\}$), the table for t can be computed if the table for t' is known. Note that $G[t]$ is the subgraph of G induced by $V(G[t']) \cup \{v\}$.

Similarly, show how, if $t \in T$ is a forget node (t has one child $t' \in T$ and there is exactly one $v \in V$ for which $X_t = X_{t'} \setminus \{v\}$), the table for t can be computed if the table for t' is known. Note that $G[t] = G[t']$.

Finally, if $t \in T$ is a join node (t has two children l and r and $X_t = X_l = X_r$), show how the table for t can be computed from the tables for l and r. Note that $G[t]$ is obtained as the disjoint union of the graphs $G[l]$ and $G[r]$ while identifying the vertices in X_t with each other.

Conclude how the size of a minimum dominating set of G may be found.

9. The MAX-SCHEDULE problem is as follows: given are m machines, n jobs, and for every $1 \leq i \leq m$ and $1 \leq j \leq n$ an integer $p_{i,j} \in \mathbb{N}_{\geq 0}$ (given in binary representation) indicating the processing time used by machine i to process job j. Additionally given is a deadline D and an integer k. The question is whether one can allocate at least k jobs to the machines such that no machine uses more than D processing time.[1]

   (a) Show how to solve this problem in polynomial time if $m = 1$.

   (b) Give an algorithm for MAX-SCHEDULE that runs in time $O^*(m^k)$. Your algorithm may have constant one-sided error probability in the following sense:

      - if the instance is a NO-instance, your algorithm should return NO,
      - otherwise, your algorithm returns YES with probability at least 1/10.

[1] More formally stated, if we denote $M_i \subseteq \{1, \dots, n\}$ for the set of jobs assigned to machine i in such an allocation, it is asked whether there exist disjoint subsets $M_1, \dots, M_m \subseteq \{1, \dots, n\}$ such that $\sum_{i=1}^{m} |M_i| \geq k$ and $\sum_{j \in M_i} p_{i,j} \leq D$ for every i.