A *path decomposition* of a graph (V, E) is a tree decomposition where the tree is a path, i.e., where each tree node has at most one child. We can then just list the sequence of bags: $(V_1, V_2, \ldots, V_r)$. A path decomposition is *nice* when $|V_1| = |V_r| = 1$ and at every next bag exactly one vertex is either added or removed, i.e., for every $i \in \{1, 2, \ldots, r-1\}$ there is a vertex $v \in V$ such that either $V_{i+1} = V_i \cup \{v\}$ or $V_{i+1} = V_i \setminus \{v\}$.

Let a graph $G = (V, E)$ be given with positive weights $w(e)$ for every edge $e \in E$ together with a nice path decomposition of G of width k. For any subset X of V the *weight* of that subset is the sum of the weights of edges (u, v) with $u \in X$ and $v \notin X$. The goal is to find the maximum weight possible over all subsets X, i.e.,

$$\max_{X \subseteq V} \sum_{(u,v) \in \{(u,v) \in E | u \in X, v \notin X\}} w(u, v).$$

Hint: use the notation $w_i(A, B)$ to denote the maximum total weight of edges between points from $\bigcup_{1 \leq j \leq i} V_j$ that are in different sets, when the partition is made consistent with a partition (A, B) of V_i.

1. (1 point) Give the maximum total weight over all partitions (A, B) of V_1 and explain.
2. (3 points) For $1 \leq i < r$, express the solution to $i+1$ in terms of those for i in case $V_{i+1} = V_i \cup \{v\}$ and argue why this is correct.
3. (2 points) For $1 \leq i < r$, express the solution to $i+1$ in terms of those for i in case $V_{i+1} = V_i \setminus \{v\}$ and argue why this is correct.
4. (2 points) Using the above recursive formulations, give the pseudocode of an iterative dynamic programming algorithm. As always, give the algorithm that uses the least space and run time.

   Hint: Compute $w_i(A, B)$ for all partitions (A, B) of V_i.
5. (1 point) What is a tight upper bound on the space required by this dynamic programming algorithm?
6. (1 point) What is a tight upper bound on the run-time complexity of this dynamic programming algorithm?