Assignment 8

CSCI 4113/6101
Instructor: Norbert Zeh
Due: Dec 9, 2025, 11:59pm

In this assignment, you are asked to have some fun with dynamic programming.

An *independent set* in a graph $G = (V, E)$ is a subset of vertices $I \subseteq V$ such that no edge in E has both its endpoints in I. Now assume that we are given a weight function $w : V \to \mathbb{R}$. Our goal is to find an independent set of maximum weight $w(I) = \sum_{v \in I} w(v)$. Even if every vertex has weight 1, that is, if our goal is to maximize the size of I, this problem is not FPT when parameterized by the size of I, under standard complexity-theoretic assumptions. Other parameterizations lead to FPT algorithms though. For example, if our aim is to maximize the size of I, then the problem is FPT when parameterized by the size of a minimum vertex cover. I leave it as an exercise for you to puzzle out why this is the case. Here, we look at a much more fun parameterization, one that allows us to use dynamic programming to obtain an FPT algorithm for the weighted version of this problem. We assume that G is connected because a maximum-weight independent set of a disconnected graph is simply the union of maximum-weight independent sets of its connected components, so solving the problem on connected graphs is the hard part.

Given an ordering $\sigma = \langle v_1, \ldots, v_n \rangle$ of the vertices of G, let $N_G^\sigma(v)$ be the set of neighbours of v that come before v in σ. Now assume that there exists a supergraph $H \supseteq G$ with the same vertex set as G (i.e., H may have additional edges but no additional vertices) such that, for every vertex $v \in V$, $|N_H^\sigma(v)| \le k$ and $H\left[N_H^\sigma(v)\right]$ is a clique. Such a supergraph always exists: Choosing H to be the complete graph with vertex set V does the trick, but this means that $k = n - 1$. The smallest value k for which such a supergraph $H \supseteq G$ exists depends on G and on the chosen ordering σ of the vertices. Finding this smallest value is another NP-hard problem that is FPT when parameterized by k, but the algorithm that does this is well beyond the scope of this course. Here, we assume that we are given a tuple (G, H, w, σ) such that $H\left[N_H^\sigma(v)\right]$ is a clique, for all $v \in V$. The value $k = \max_{v \in V} \left|N_H^\sigma(v)\right|$ is then easy to compute.

The goal of this assignment is to show that, given such a tuple (G, H, w, σ), a maximum-weight independent set in G can be found in $O\left(4^k n + 2^k nm\right)$ time.

As a preprocessing step, we need to construct a rooted tree T that represents G. The vertex set of T is V, the vertex set of G. Referring to the vertices of G by their indices in $\sigma = \langle v_1, \ldots, v_n \rangle$, the root of T is v_1. For $i > 1$, the parent of v_i is the vertex $v_j \in N_H^\sigma(v)$ with maximum index j. This tree T is easy to construct in $O(kn)$ time from H and σ (the construction takes $O(|E(H)|)$ time, and it is not hard to show that H has at most kn edges). Question (1) below asks you to show that this tree is well-defined, that is, that $N_H^\sigma(v_i)$ is non-empty, for all $i > 1$.

Now, for all $i \in [n]$, let X_i be the set of descendants of v_i in T, including v_i itself, and let $G_i = G\left[N_H^\sigma[v_i] \cup X_i\right]$. Your goal is to construct a dynamic programming table I indexed by pairs (i, U), where $i \in [n]$ and $U \subseteq N_H^\sigma[v_i]$. For any such pair, (i, U), let $\mathcal{I}_{i,U}$ be the set of all independent sets I of G_i with $I \cap N_H^\sigma[v_i] = U$. For each independent set $I \in \mathcal{I}_{i,U}$, let $w_i(I) = w(I \cap X_i)$ be the total weight of all vertices in $I \cap X_i$. Then $I[i, U] = \max\{w_i(I) \mid I \in \mathcal{I}_{i,U}\}$.

(1) Prove that $N_H^\sigma(v_i) \neq \emptyset$, for all $i > 1$.

(2) Prove that $G_1 = G$ and that this implies that the weight of a maximum-weight independent set in G can be computed from the table entries in I in constant time.

(3) Prove that, if v_i is a leaf of T, then the table entry $I[i,U]$, for each $U \subseteq N_H^\sigma[v_i]$, can be computed in $O(k+m)$ time.

(4) Prove that, if v_i is a non-leaf vertex of T with $d \geq 1$ children $v_{j_1}, \dots, v_{j_d}$, then the table entry $I[i,U]$ can be computed from appropriate table entries $I[j_h, U_h]$, where $h \in [d]$ and U_h is an appropriate subset of $N_H^\sigma\left[v_{j_h}\right]$. There are cases when you have to consider multiple choices of U_h for each child v_{j_h} of v_i. Prove that computing each table entry $I[i,U]$ takes $O\left(2^k d + m\right)$ time.

(5) Combine your answers to (2)–(4) to show that a maximum-weight independent set in G can be found in $O\left(4^k n + 2^k nm\right)$ time.

Hint: *The main reason why it is possible to solve this problem in FPT time parameterized by k is that the vertices in $N_H^\sigma[v_i]$ form a "separator" between G_i and the rest of G in the sense that every edge $\{u,v\} \in G$ with $u \in G_i$ and $v \notin G_i$ satisfies $u \in N_H^\sigma[v_i]$. To gain some intuition about the structural information about G provided by H and σ—and, by extension, by T—you should prove this claim.*

You should then use this fact to show that each table entry $I[i,U]$, for a non-root vertex v_i, can be computed by considering an appropriate subset of those table entries $I[j_h, U_h]$ associated with each of its children v_{j_h} in T that satisfy $U_h \supseteq U \cap N_H^\sigma\left[v_{j_h}\right]$.

Marking Scheme

No more marking scheme. Submit this for formative assessment only, but do give it your best shot and try to have fun.

Submission Instructions

Follow the submission link for this assignment on the course webpage in the email you should have received from Crowdmark. Upload the assignment as a single PDF file.