CSE515: Advanced Algorithms
Notes on Lecture 17: Greedy Algorithm for Set Cover

Antoine Vigneron

April 27, 2021

We prove here that SET COVER is **NP**-hard. As we already know that VERTEX COVER is **NP**-hard, we use a reduction from VERTEX COVER, i.e. we prove that VERTEX COVER $\leqslant_p$ SET COVER.

So suppose that we have an algorithm A that solves SET COVER in polynomial time. We will show that it allows us to solve VERTEX COVER in polynomial time.

Let $G(V, E)$ be an instance of vertex cover, with $V = \{v_1, \ldots, v_m\}$ and $E = \{e_1, \ldots, e_n\}$. (Remark: we use m for the number of vertices and n for the number of edges, while usually n is the number of vertices and m the number of edges.)

We now construct an instance of SET COVER by setting $U = \{e_1, \ldots, e_n\}$, and letting S_i be the set of edges e_j incident to v_i in G, that is,

$$S_i = \{e_j \in E \mid e_j = (v_i, w) \text{ for some } w \in V\}$$

So the vertex v_i covers the edge e_j in G if and only if the set S_i covers e_j seen as an element of U. Finally, we set $w_i = 1$.

A subset of vertices $C = \{v_{i_1}, \ldots, v_{i_k}\}$ is thus a vertex cover of G of size k if and only if $\{S_{i_1}, \ldots, S_{i_k}\}$ is a set cover of weight k for U. Hence, in order to solve our instance of VERTEX COVER, it suffices to compute a minimum set cover $S_{i_1}, \ldots, S_{i_k}$ of U using algorithm A, and return the vertex cover $v_{i_1}, \ldots, v_{i_k}$ of G. Therefore we have solved VERTEX COVER in polynomial time given a SET COVER algorithm A.