Section Handout 8

Problem One: NP

For each of the following languages, show that the language is in **NP** by designing a polynomial-time verifier.

i. Given a sequence of numbers $x_1, x_2, \ldots, x_n$, an ***ascending subsequence*** is a subsequence of the original sequence (that is, some elements of the sequence taken in the original order in which they appear) such that each term is larger than the previous term. For example, given the sequence 2, 3, 0, 1, 4, the subsequence **2**, **3**, **4** is an ascending subsequence, as is **0**, **1**, **4**. Let $ASCEND = \{ \langle x_1, x_2, \ldots, x_n, k \rangle \mid$ There is an ascending subsequence of $x_1 \ldots x_n$ with length at least k. $\}$ Prove that $ASCEND \in$ **NP** by designing a polynomial-time verifier for it.

ii. In an undirected graph $G = (V, E)$, a ***dominating set*** is a set $D \subseteq V$ such that every node in V either belongs to D or is connected to a node in D by an edge. Every graph has a dominating set consisting of every node in the graph, though it's unclear whether smaller dominating sets exist.

   Let $DS = \{ \langle G, k \rangle \mid G$ is an undirected graph containing a dominating set with at most k nodes $\}$. Prove that $DS \in$ **NP** by designing a polynomial-time verifier for it. (It turns out that $DS \in$ **NPC** as well, though that proof is a bit harder.)

Problem Two: NP-Completeness

The independent set problem, as covered in lecture, is specified as follows:

$$INDSET = \{ \langle G, k \rangle \mid G \text{ is an undirected graph that contains an independent set of size } k \}$$

As we saw in lecture, *INDSET* is **NP**-complete. Using the fact that *INDSET* is **NP**-complete, you will prove that the **set packing problem** is **NP**-complete as well.

In the set packing problem, you are given a list of n sets $S_1, S_2, \ldots, S_n$ along with a number k. The goal is to answer the question

Is there a collection of k sets from the list $S_1, S_2, \ldots, S_n$
such that no element is contained in two of those k sets?

For example, given the sets

$$\{1, 3, 5\}, \{1, 2, 3\}, \{2, 4\}, \{2, 5, 7\}, \{6\}$$

And the number 3, we would answer "yes" because the collection of sets $\{1, 3, 5\}$, $\{2, 4\}$, and $\{6\}$ collectively have no elements in common with one another.

Formally, we define the set packing problem as

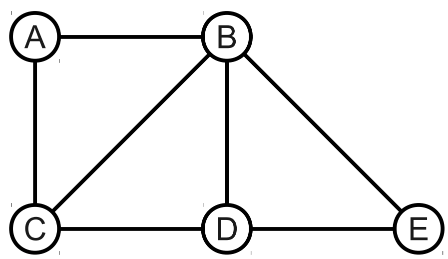
$$SETPACK = \{ \langle S_1, S_2, \ldots, S_n, k \rangle \mid \text{There are } k \text{ mutually non-overlapping sets in } S_1 \ldots S_n \}$$

i. Prove that $SETPACK \in \mathbf{NP}$ by designing an NTM that decides it in polynomial time.

To show that SETPACK is **NP**-complete, we will reduce the *INDSET* problem to it. Given a graph $G = (V, E)$, we will construct a family of sets whose elements are the edges in G. There will be one set for each vertex in the graph. Specifically, for each node in $v_i \in V$, we will create a set S_i defined as follows:

$$S_i = \{ \{v_i, v_j\} \mid \{v_i, v_j\} \in E \}$$

That is, the set associated with the vertex v_i is the set of all edges incident to v_i. For example, given this graph:



We would construct the sets

- $S_A = \{ \{A, B\}, \{A, C\} \}$
- $S_B = \{ \{A, B\}, \{B, C\}, \{B, D\}, \{B, E\} \}$
- $S_C = \{ \{A, C\}, \{B, C\}, \{C, D\} \}$
- $S_D = \{ \{B, D\}, \{C, D\}, \{D, E\} \}$
- $S_E = \{ \{B, E\}, \{D, E\} \}$

ii. Using this reduction, prove that $SETPACK \in \mathbf{NPC}$ by showing $INDSET \leq_P SETPACK$.

Thanks for attending! Good luck on the final exam!