

## CS302 - Problem Set 11

Recall the following definitions:

- **$k$ -INDSET:** Given an undirected, unweighted graph  $G = (V, E)$ , is there a set  $V' \subseteq V$  such that  $|V'| \geq k$ , and for all  $v, u \in V'$ , there is no edge  $\{u, v\} \in E$ ? We call such a set a  $k$ -independent set.
- **$k$ -CLIQUE:** Given an undirected, unweighted graph  $G = (V, E)$ , is there a set  $V' \subseteq V$  such that  $|V'| \geq k$ , and for all  $v, u \in V'$ , there is an edge  $\{u, v\} \in E$ ? We call such a set a  $k$ -clique.
- **DOUBLE-3-SAT:** Given a CNF formula with at most  $m$  clauses, where  $m$  is a polynomial, and each clause involves at most 3 of the variables  $x_1, x_2, \dots, x_n$  and their negations, are there at least two different satisfying solutions? For example,  $(x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_3)$  has two valid assignments,  $x_1 = 1, x_2 = 1, x_3 = 0$  and  $x_1 = 0, x_2 = 1, x_3 = 0$ .

In the following, you can assume the results you proved in the prior PSet, that  $k$ -INDSET,  $k$ -CLIQUE, and DOUBLE-3-SAT are all in NP.

1. Prove DOUBLE-3-SAT is NP-Complete.
2. Prove that  $k$ -INDSET reduces to  $k$ -CLIQUE.
3. Prove that 3-SAT reduces to  $m$ -INDSET, where  $m$  is the number of clauses in the 3-SAT.
4. Prove that  $k$ -INDSET and  $k$ -CLIQUE are NP-Complete. (This is a short proof!)
5. *Note: This problem will provide good practice in designing a dynamic programming algorithm start to finish. This will likely not be an easy problem for many of you - we have not seen a problem with strings like this before. The hardest part will probably be figuring out the relevant subproblems and recurrence. Please struggle with this before looking at the hints.)*

Given two strings  $x$  and  $y$ , the edit distance  $D(x, y)$  is the minimum number of insertions or deletions or substitutions (a substitution involves replacing one character in the string with another) required to turn  $x$  into  $y$  (or vice versa). This is used for spell checkers: if someone types a word that is not in the dictionary, you want to find the word that is closest to it in edit distance. For example, if someone typed “graffe” its edit distance from “graft” is 2 (delete “e”, substitute “f” for “t”), while its edit distance to “giraffe” is 1 (insert “i”).

Create a dynamic programming algorithm that takes in two strings ( $x$  and  $y$ ), and calculates  $D(x, y)$ . Think of options for the final alteration, define subproblems, create a recurrence, and then write pseudocode to find the minimum edit distance.

6. Approximately how long did you spend on this assignment (round to the nearest hour)?

**Hints:**

1. Show how to reduce 3-SAT to Double-3-Sat.
2. Look up the *complement* of a graph. It will be helpful.
3. Try to come up with a simple graph gadget for each clause. Then figure out how to connect the gadgets.
- 5.

Hint 1: If  $x$  and  $y$  don't have the same final letter, then you can think about changing (adding to, subtracting, substituting) the final character in  $x$ . If  $x$  and  $y$  have the same final letter, then you only need to work on fixing the letters up until the final character.

Hint 2: Relevant subproblems: Let  $x_i$  be the  $i$ th character in  $x$ , and let  $\vec{x}_i$  be the first  $i$  letters of  $x$ . Likewise for  $y$ . Let  $T(\vec{x}_n, \vec{y}_m)$  be the set of optimal transformations to turn  $\vec{x}_n$  into  $\vec{y}_m$ .

Hint 3:

$$T(\vec{x}_n, \vec{y}_m) = \begin{cases} \text{????} & \text{if } x_n = y_m \\ \text{????} & \text{if last transformation is delete } x_n \\ \text{????} & \text{if last transformation is add } y_m \text{ to } \vec{x}_n. \\ \text{????} & \text{if last transformation is sub } x_n \text{ with } y_m \end{cases} \quad (1)$$

Hint 4: Your base cases should include the first row and the first column of the array. Why? What should you fill in with?