

CS302 - Problem Set 11

1. We've been learning about frameworks for deciding when algorithms are problematic and mitigating their harm. However, these frameworks are fairly abstract, and not yet in common use, and don't necessarily deal with the real world experience of discovering that your company or another company's algorithm is causing harm, and then working to mitigate that harm. We will look at two case studies of people who attempted to alert companies to harm or potential harm due their algorithms, and the results of their efforts:

- Watch this Youtube video on the research of Joy Buolamwini. Then for a recent update on this story, read the first few paragraphs of IBM Abandons Facial Recognition (NPR). Optional: for a more in-depth look at Dr. Buolamwini's research and story and facial recognition more broadly, you should watch the movie Coded Bias (will need Midd credentials to log in.)
- Read about Timnit Gebru's Firing from Google (Washington Post)

Comment on Dr. Buolamwini's approach to interacting with companies she studied. Which of her strategies were more successful in accomplishing change, and why do you think that was? Comment on how different companies reacted differently to her research.

Comment on Google's treatment of Dr. Gebru, and their approach to ethical research more generally (at least as described in the article). What are your feelings/reactions? What are some of the benefits and harms of Google's (in)actions on ethics? (I'm assuming there are some benefits, or Google wouldn't act the way it does!)

2. Recall the following definitions:

- k -INDSET: Given an undirected, unweighted graph $G = (V, E)$, output YES if there is 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.) Otherwise output NO.
- k -CLIQUE: Given an undirected, unweighted graph $G = (V, E)$, output YES if there is 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.) Otherwise output NO.
- DOUBLE-3-SAT: Given a CNF formula with n variables, where each clause involves at most 3 of the variables z_1, z_2, \dots, z_n (and their negations) output YES if there at least two *different* satisfying assignments to the formula. Otherwise output NO. For example, $(z_1 \vee \neg z_1 \vee \neg z_2) \wedge (z_2 \vee z_3) \wedge (\neg z_3)$ is a YES instance because it has two valid assignments, $z_1 = 1, z_2 = 1, z_3 = 0$ and $z_1 = 0, z_2 = 1, z_3 = 0$.

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.

- (a) Prove DOUBLE-3-SAT is NP-Complete.
- (b) In this problem, you'll prove that that k -INDSET and k -CLIQUE are NP-Complete.
 - i. Prove that there is a polynomial reduction from k -INDSET to k -CLIQUE.
 - ii. Prove that there is a polynomial reduction from 3-SAT to m -INDSET, where m is the number of clauses in the 3-SAT.
 - iii. Use the previous two parts to prove that k -INDSET and k -CLIQUE are NP-Complete.
- (c) **Challenge:** Consider the problem of outputting the *second* largest value of an array A . A is an unsorted array A of unique integers of length n (you may assume n is a power of 2).
 - i. Describe an algorithm that uses exactly $n + \log_2 n - 2$ comparison operations, where a comparison is an operation that tests whether one element is less than another element.
 - ii. Justify why your algorithm is correct. (You can do this as a proof, or as a slightly less formal explanation.)

(Hint: the algorithm combines ideas from both divide and conquer and dynamic programming.)