Ella Neurohr, Chidera Ikpeamarom, Jorge Madrigal Adan

Worksheet #12 Complexity

1. Define the complexity class, P. Give an example of an algorithm you have written that is in P.

The complexity class P is the set of problems that take a deterministic machine polynomial time to solve. An example of this would be quick-sort.

2. Define the complexity class, NP. Give an example of an algorithm you have written that is in NP.

The complexity class NP is the set of problems that can be solved by a nondeterministic machine in polynomial time. The answers to an NP problem can be verified in polynomial time. An example of an NP-hard algorithm is the Traveling Salesman problem.

3. Define the P = NP question. What is known and what is not known? What are the implications of a result?

The question of P=NP is if NP-hard problems could be reduced to P problems, and therefore solved in polynomial time. We do not know if P and NP are the same. The implications of finding this to be true not only include becoming wildly famous and rich enough to pay off our student debt, but also that for a problem which takes a non-deterministic machine polynomial time to solve a problem could be built as a deterministic machine that could solve the same problem in polynomial time.

4. What does the syntax $A \leq_P B$ mean?

Given two languages, A and B, A is polynomial time reducible to B if a polynomial time function exists. The input of A can be transformed to the input of B.

$A \leq_P B$: $w \in A$ iff $f(w) \in B$

5. Are the following two statements true? Why or why not? $A \leq_P A$ if $A \leq_P B$ and $B \leq_P C$, then $A \leq_P C$

It is true. If A is polynomial time reducible to B which is polynomial time reducible to C, then A is reducible to C in polynomial time. If the input of A can be transformed to the input of B where B solves a subproblem in polynomial time, and again from B to C, that would be the addition of two polynomial time reductions, which is still polynomial time.

6. Is the following list of numbers an element of the sum of subset language? [6, 3, 12, -1, -5, 4, 17]

The entire list isn't an element of the sum of the subset language itself since it doesn't add up to equal 0, but there is a singular subset of this list that sums to 0: {6, -1, -5}.

7. Is the following formula an element of 3SAT? If so, what variable values satisfy it?

(x0 ∨ !x2 ∨ x3) ∧ (x1 ∨ x2 ∨ !x3) ∧ (!x0 ∨ x2 ∨ !x4)

Yes,

x0: TRUE
x1: FALSE
x2: TRUE
x3: FALSE
x4: FALSE

8. Suppose you can solve a certain kind of problem with a deterministic TM in polynomial time. What does this tell you about the amount of space you would need to solve the problem?

The amount of space required would not surpass the time complexity of the problem. With each step at most one new cell can be used, so no more cells than timesteps can be used. It could be less since cells can be written over as well.