

## Question 2

### a) Polynomial-time algorithm to solve DNF-SAT

In DNF, only one of the groups of and clauses has to be true. Within each group of ANDs the only case when it isn't satisfiable is when both a variable and it's negation are present. Thus, you can just iterate through all of the groups, and for each group use a map to store which variables are present and whether they're negated or not. If you find two literals that are the negation of the other, return FALSE. Otherwise return TRUE.

### b) What's the error in the argument that P=NP?

This argument assumes that you can transform any 3SAT problem into CNF in polynomial time. This is likely not the case, since adding more clauses can exponentially increase the number of elements in CNF form.

For just two sets of two variables we can see that it expands to double the original number of clauses.

$$(x \vee y) \wedge (x \vee y) \iff (x \wedge x) \vee (x \wedge y) \vee (y \wedge x) \vee (y \wedge y)$$

For three sets of two, there's 8 sets of two created. This causes an exponential size of the expansion.