1. (25 points) Assume \( L_1 \in DTIME(T_1(n)) \) and \( L_2 \in DTIME(T_2(n)) \). Show that \( L_1 \cup L_2 \in DTIME(T_1(n) + T_2(n)) \). (You can write pseudo code and note how long the program runs. We ignore multiplicative and additive constants.)

**SOLUTION TO PROBLEM ONE**

\( L_1 \in DTIME(T_1(n)) \) via \( M_1 \), \( L_2 \in DTIME(T_2(n)) \) via \( M_2 \).

(a) Input(\( x \)) of length \( n \).

(b) Run \( M_1(x) \). If it returns YES then output YES and halt.

(c) Run \( M_2(x) \). If it returns YES then output YES and halt.

(d) If you get here then return NO.

Most time spent was in running \( M_1(x) \), which takes \( \leq T_1(n) \) steps, and in running \( M_2(x) \), which takes \( \leq T_2(n) \) steps. Hence the total time is \( \leq T_1(n) + T_2(n) \).

2. (25 points) Formally (using tuple notation) define a 2-dimensional Turing machine with a single halt state. Its input will be an rectangle of symbols. Also, briefly describe the transition function.

**SOLUTION TO PROBLEM TWO**

\((Q, \Sigma, \delta, s, h)\)

(a) \( Q \) is a set of states.

(b) \( \Sigma \) is an alphabet.

(c) \( s \in Q \) is the start state.

(d) \( h \in Q \) is the halt state - once there you are DONE.

\[ \delta : (Q - \{h\}) \times \Sigma \rightarrow Q \times (\{R, L, U, D\} \cup \Sigma) \]

The intuition is that on input a symbol it can go left, right, up, down, or overwrite with an element of \( \Sigma \).
3. (25 points) Let $L \in DTIME(T(n))$. Find polynomials $p$ such that $L^* \in DTIME(p(n)T(n))$. Give an algorithm that achieves this (it can use the algorithm for $L \in DTIME(T(n))$ and should be in pseudocode).

**SOLUTION TO PROBLEM THREE**

$p(n) = n^2$. See the algorithm below.

```plaintext
input x = x_1 \ldots x_n
set A[0] = TRUE
# A[k] denotes whether (x_1 \ldots x_k) is in L^*
for j = 1 to n do
    for k = 0 to j-1 do
        if A[k] and (x_{k+1} \ldots x_j) is in L then
            A[j] = TRUE
        end
    end
end
output A[n]
```

GO TO NEXT PAGE
4. A formula is in *DNF-form* if it is of the form

\[ D_1 \lor D_2 \lor \cdots \lor D_L \]

where each \( D_i \) is a \( \land \) of literals. (DNF stands for Disjunctive Normal Form.) We call the \( d \)'s **DISJUNCTS**.

(a) (10 points) Show that the following problem is in \( P \): given a formula in DNF form, determine if it is satisfiable.

(b) (8 points) Write the following CNF formula in DNF form:

\[ \phi_3 = (x_1 \lor y_1) \land (x_2 \lor y_2) \land (x_3 \lor y_3) \]

How many disjuncts are in your formula?

(c) (7 points) Write the following CNF formula in DNF form (you can describe how you would do it, but be clear).

\[ \phi_n = (x_1 \lor y_1) \land (x_2 \lor y_2) \land \cdots \land (x_n \lor y_n) \]

How many disjuncts are in your formula?

(d) (0 points but think about, DO NOT hand anything in for this part) Your answer to (c) should be a large function, NOT a polynomial. This means that YOUR attempt to get this CNF formula into a DNF formula causes a blowup in size. I will ask the class to vote for either

- There is a poly-sized DNF formula for \( \phi_n \) AND this is known.
- There is NO poly-sized DNF formula for \( \phi_n \) AND this is known.
- Whether or not there is a poly-sized DNF formula for \( \phi_n \) is UNKNOWN TO SCIENCE!

**SOLUTION TO PROBLEM FOUR**

a) Poly time algorithm.

Given \( D_1 \lor \cdots \lor D_L \) do the following

For \( 1 \leq i \leq L \) look at \( D_i \). If \( D_i \) does not have both a var and its negation in it, then \( D_i \) can be satisfied, so the entire formula can be satisfied, so output YES and stop.
If you have gone through every $D_i$ and not found that, then output NO and stop.

b) 

$$\phi_3 = (x_1 \lor y_1) \land (x_2 \lor y_2) \land (x_3 \lor y_3)$$

For this to be satisfied we need at least one from each clause to be satisfied. So we need to list out all the ways that can happen.

$$(x_1 \land x_2 \land x_3) \lor$$

$$(x_1 \land x_2 \land y_3) \lor$$

$$(x_1 \land y_2 \land x_3) \lor$$

$$(x_1 \land y_2 \land y_3) \lor$$

$$(y_1 \land x_2 \land x_3) \lor$$

$$(y_1 \land x_2 \land y_3) \lor$$

$$(y_1 \land y_2 \land x_3) \lor$$

$$(y_1 \land y_2 \land y_3)$$

There are 8 disjuncts.

c) 

$$\phi_n = (x_1 \lor y_1) \land (x_2 \lor y_2) \land \cdots \land (x_n \lor y_n)$$

The DNF for this is a $\lor$ of ALL possible ways to pick one variable from each clause. For every $z \in \{0, 1\}^n$ we have the disjunct that has

$x_i$ if $z_i = 1$

$y_i$ if $z_i = 0$

So it has $2^n$ disjuncts.