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.)

2. (25 points) Formally define a 2-dimentional Turing machine. Its input will be an rectangle of symbols.

3. (25 points) Let $L \in DTIME(T(n))$. Find a polynomial $p$ such that $L^* \in DTIME(p(T(n))$. Give the algorithm that achieves this (it can use the algorithm for $L \in DTIME(T(n))$ and should be in pseudocode).

GO TO NEXT PAGE
4. A formula is in \textit{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 point but think about, DO NOT hand anything in for this part)

Your answer to 7 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 \textit{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 \textit{UNKNOWN TO SCIENCE}!