We study TM's to compare it against algorithm's running time.
(Like horse power)

From the homework:
Method to emulate a TM:

```java
public static void cpu(int N, int M, int[] state, int[] [] symbol, char[] str, int [] tape) {
    int location = 0;
    int head = 0; // in state
    int cSymbol;
    int cState;

    while (head < N-2) {  
        cSymbol = tape[location];
        tape[location] = symbol[head][cSymbol];
        cState = head;
        head = state[head][cSymbol];
        if (cState == cSymbol) 
            location++;
        else if (cState == cSymbol) 
            location--;
    }
}
```
A Turing Machine is a finite automaton with 2 stacks

EA - the head

1st stack: where the head is to the right
2nd stack: to the left of the head

Example: check if string ends with 'a'.

What is a “problem”?

Example 1: mathematics

given: a statement x
We want a proof $y$ of either $x$ or $\neg x$.

There is a feasible algorithm $c(x,y)$ such that $\text{len}(y) \leq P_e(\text{len}(x))$.

**Given:** $x$ Find $y$ such that $c(x,y)$ holds and $\text{len}(y) \leq P_e(\text{len}(x))$.

**General Problem:** $c(x,y)$ feasible algorithm and a polynomial $P_e(n)$.

$$c(x,y) = \begin{cases} \text{true} & \text{if } y \text{ is a proof of } x \text{ or } \neg x \\ \text{false} & \text{otherwise} \end{cases}$$

**Physics:** Given data $x$ Find a "law" $y$ that explains the data

**Example**

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

Checking: That $y$ is consistent with $x$ is feasible $c(x,y)$

$\text{len}(y) \leq \text{len}(x)$ \quad $P_e(n) \leq n$
should be less

Engineering: Given: Task X (specification) $c(x,y)$-feasible algo.
Find: Design $y$
$\text{len}(y) \leq P_e(\text{len}(x))$

Difference between the three:
Given: $x$ task
Music: $c(x,y)$ is not algorithmic

A problem is:
A pair consisting of $(c, P_e)$ where $C$ is a feasible algorithm and $P_e$ is a polynomial

An instance is:
Given $x$, find $y$ such that $c(x,y)$ is true and $\text{len}(y) \leq P_e(\text{len}(x))$

Class of all problems is called NP.
Non-deterministic Polynomial

$\mathbb{P}$ is the class of all problems that can be solved using a feasible algorithm.