Def: DTIME, NTIME, DSPACE, measured on Multi-tape Turing Machines.

Th: DTIME[t(n)] ⊆ RAM-TIME[t(n)] ⊆ DTIME[(t(n))^3]

\[ L \equiv \text{DSPACE}[\log n] \]
\[ P \equiv \text{DTIME}[n^{O(1)}] \equiv \bigcup_{i=1}^{\infty} \text{DTIME}[n^i] \]
\[ \text{NP} \equiv \text{NTIME}[n^{O(1)}] \equiv \bigcup_{i=1}^{\infty} \text{NTIME}[n^i] \]
\[ \text{PSPACE} \equiv \text{DSPACE}[n^{O(1)}] \equiv \bigcup_{i=1}^{\infty} \text{DSPACE}[n^i] \]

Th: For \( t(n) \geq n, s(n) \geq \log n, \)

\[ \text{DTIME}[t(n)] \subseteq \text{NTIME}[t(n)] \subseteq \text{DSPACE}[t(n)] \]
\[ \text{DSPACE}[s(n)] \subseteq \text{DTIME}[2^{O(s(n))}] \]

Cor: \( L \subseteq P \subseteq \text{NP} \subseteq \text{PSPACE} \)
**Definition 6.1** The *busy beaver function* $\sigma(n)$ is the maximum number of one’s that an $n$ state TM with alphabet $\Sigma = \{0, 1\}$ can leave on its tape and halt when started on the all 0 tape. (To fit our definitions, note that “0” is now the “blank character”.)

Note that $\sigma(n)$ is well defined:
There are only finitely many $n$-state TMs, with $\Sigma = \{0, 1\}$. Some finite subset, $F_n$, of these eventually halt on input 0.
Some element of $F_n$ prints the max # of 1’s $= \sigma(n)$. 

♠
<table>
<thead>
<tr>
<th></th>
<th>$q_1$</th>
<th>$q_2$</th>
<th>$q_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$q_2, 1,\rightarrow$</td>
<td>$q_3, 0,\rightarrow$</td>
<td>$q_3, 1,\leftarrow$</td>
</tr>
<tr>
<td>1</td>
<td>$h, 1,\neg$</td>
<td>$q_2, 1,\rightarrow$</td>
<td>$q_1, 1,\leftarrow$</td>
</tr>
</tbody>
</table>

\[
\sigma(3) \geq 6
\]
How quickly does $\sigma(n)$ grow as $n$ gets large?

Is $\sigma(n) \in O(n^2)$ ?

$O(n^3)$ ?

$O(2^n)$ ?

$O(n!)$ ?

$O(2^{2^n})$ ?

$O(\exp^*(n))$ ?

$O(\exp^*(\exp^*(n)))$ ?

\[
\begin{align*}
\exp^*(n) &= 2^{\left\lfloor 2^{\left\lfloor 2^{\cdots^{2}}\right\rfloor} \right\rfloor} \\
&= 2^{\left\lfloor 2^{\left\lfloor 2^{\cdots^{2}}\right\rfloor} \right\rfloor}
\end{align*}
\]
<table>
<thead>
<tr>
<th>States</th>
<th>Max # of 1’s</th>
<th>Lower Bound for $\sigma(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>$\sigma(3)$</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>$\sigma(4)$</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>$\sigma(5)$</td>
<td>$\geq 4098$</td>
</tr>
<tr>
<td>6</td>
<td>$\sigma(6)$</td>
<td>$&gt; 10^{865}$</td>
</tr>
</tbody>
</table>

See the web pages of Penousal Machado (eden.dei.uc.pt) and Heiner Marxen (www.drb.insel.de/heiner/BB) for more on this problem and its variants.
Theorem 6.2 Let $f : \mathbb{N} \to \mathbb{N}$ be a total, recursive function.

$$\lim_{n \to \infty} \left( \frac{f(n)}{\sigma(n)} \right) = 0$$

That is, $f(n) = o(\sigma(n))$.

Proof:

$$g(n) = n \cdot \left( 1 + \sum_{i=0}^{n} f(i) \right)$$

Note:

$$\lim_{n \to \infty} \left( \frac{f(n)}{g(n)} \right) = 0$$

We will show for all sufficiently large $n$,

$$\sigma(n) \geq g(n)$$
$g(n)$ is computed by a $k$-state TM for some $k$.
For any $n$, define the TM

$$C_n = \begin{array}{c}
\text{print } n \\
\text{compute } g \\
\text{binary to unary}
\end{array}
\begin{array}{c}
\lceil \log n \rceil \\
k \\
17
\end{array}$$

$C_n$ has $\lceil \log n \rceil + k + 17$ states.
$C_n$ prints $g(n)$ 1’s.
Once $n$ is big enough that $n \geq \lceil \log n \rceil + k + 17$,
$$\sigma(n) \geq \sigma(\lceil \log n \rceil + k + 17) \geq g(n)$$
On HW#2, we define a pairing function:

$$P : \mathbb{N} \times \mathbb{N} \xrightarrow{1:1} \mathbb{N}$$

$$P(L(w), R(w)) = w$$
$$L(P(i, j)) = i$$
$$R(P(i, j)) = j$$

We can use the pairing function to think of a natural number as a pair of natural numbers.

Thus, the input to a Turing machine is a single binary string which may be thought of as a natural number, a pair of natural numbers, a triple of natural numbers, and so forth. (Later we will worry about the complexity of the pairing and string-conversion functions – do you think they are in $\mathbb{L}$)?
Turing machines can be encoded as character strings which can be encoded as binary strings which can be encoded as natural numbers.

<table>
<thead>
<tr>
<th>$\text{TM}_n$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1,0,→</td>
<td>3,□,→</td>
<td>0,0,−</td>
<td>0,0,−</td>
</tr>
<tr>
<td>1</td>
<td>1,1,→</td>
<td>4,□,→</td>
<td>0,1,−</td>
<td>0,1,−</td>
</tr>
<tr>
<td>□</td>
<td>2,□,←</td>
<td>0,□,−</td>
<td>1,0,←</td>
<td>1,1,←</td>
</tr>
<tr>
<td>▶</td>
<td>1,▶,→</td>
<td>0,▶,−</td>
<td>0,▶,−</td>
<td>0,▶,−</td>
</tr>
</tbody>
</table>

ASCII: 1,0,→; 1,1,→; 2,□,←; 1,▶,→; ··· 0,▶,−

$\{0,1\}^*: w$

$N:\ n$

There is a simple, countable listing of all TM’s:

$M_0, M_1, M_2, \cdots$
Theorem 6.3 There is a Universal Turing Machine $U$ such that,

$$U(\langle n, m \rangle) = M_n(m)$$

Proof: Given $\langle n, m \rangle$, compute $n$ and $m$. $n$ is a binary string encoding the state table of TM $M_n$. We can simulate $M_n$ on input $m$ by keeping track of its state, its tape, and looking at its state table, $n$, at each simulated step. ♠

Let’s look at $L(U)$, the set of numbers $P(n, m)$ such that the Turing machine $M_n$ eventually halts on input $n$. We’ll call this language HALT. The existence of $U$ proves that HALT is r.e., and we’ll now show it’s not recursive.
Theorem 6.4 (Unsolvability of the Halting Problem)
HALT is r.e. but not recursive.

Proof:

\[
\text{HALT} = \{w \mid U(w) \text{ eventually halts}\} = \{w \mid U'(w) = 1\}
\]

\[
U' = \begin{array}{c|c|c}
U & \text{erase tape} & \text{print 1} \\
\end{array}
\]

Suppose HALT were recursive. Then \(\sigma(n)\) would be a total recursive function: Cycle through all \(n\)-state TMs, \(M_i\), and if \(P(i, 0) \in \text{HALT}\), then count the number of 1’s in \(M_i(0)\). Return the maximum of these. But \(\sigma(n)\) isn’t total recursive, so we have a contradiction.

\[\spadesuit\]
\[ W_i = \{ n \mid M_i(n) = 1 \} \]

The set of all r.e. sets = \( W_0, W_1, W_2, \cdots \)

<table>
<thead>
<tr>
<th>( n )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>( W_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( W_0 )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( W_1 )</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>( W_2 )</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>( W_3 )</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>( W_4 )</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>( W_5 )</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>( W_6 )</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( W_7 )</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( W_8 )</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
<td>\vdots</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( K )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( \overline{K} )</td>
</tr>
</tbody>
</table>
\[ K = \{ n \mid M_n(n) = 1 \} \]
\[ = \{ n \mid U(P(n,n)) = 1 \} \]
\[ = \{ n \mid n \in W_n \} \]

**Theorem 6.5**  \( K \) is not r.e.

**Proof:**  \( \overline{K} = \{ n \mid n \notin W_n \} \)

Suppose \( \overline{K} \) were r.e. Then for some \( c \),

\[ \overline{K} = W_c = \{ n \mid M_c(n) = 1 \} \]

\[ c \in K \iff M_c(c) = 1 \iff c \in W_c \iff c \in \overline{K} \]

\( \spadesuit \)

**Corollary 6.6**  \( K \in \text{r.e.} - \text{Recursive} \)