Some facts about TMs

- Every string x represents a TM M_x, and every TM M can be represented by an infinite number of strings.
- Universal TM U simulates any TM M_x that runs in time $f(n)$ in time $O(f(n)\log f(n))$.
- Function f is time-constructible iff $f(n)$ can be computed in $O(f(n))$ time.

Theorem 1

$DTIME(n) \subset DTIME(n^{1.5})$

Proof: By contradiction.

Algorithm $D(x)$

Run $U(M_x, x)$ for $|x|^{1.4}$ steps
if $U(M_x, x)$ hasn't finished **then**
  return 0
else
  return $\neg U(M_x, x)$

Language $D \in DTIME(n^{1.5})$. Assume language $D \in DTIME(n)$, i.e., TM M decides D in $O(n)$ time.

- $U(M, x)$ runs in $O(|x| \log |x|) \leq c \cdot |x| \log |x|$ time for $c > 0$
- $\exists n_0 > 0 \; \forall n > n_0 : n^{1.4} > cn \log n$

Let $\llcorner M \lrcorner \geq n_0 \Rightarrow U(M, \llcorner M \lrcorner) = \neg M(\llcorner M \lrcorner)$ **contradiction!** $\square$

Theorem 2

$DTIME(n) \subset DTIME(\Theta(n \log n))$

Proof: By contradiction.

Algorithm $D(x)$

```
Run U(M_x, x) for Θ(|x| log |x|) steps
if U(M_x, x) hasn't finished then
    return 0
else
    return ¬U(M_x, x)
```

Language $D \in DTIME(\Theta(n \log n))$.

- Assume language $D \in DTIME(n)$, i.e., TM M decides D in $O(n)$ time.
- $U(M, x)$ runs in $O(|x| \log |x|)$ time
- $M(\llcorner M \lrcorner) = U(M, \llcorner M \lrcorner) = D(\llcorner M \lrcorner) = \neg M(\llcorner M \lrcorner)$ **contradiction!**

$\square$

Theorem 3

If f, g are time-constructible and $\lim_{n\to\infty} \frac{f(n)\log f(n)}{g(n)} = 0$, then

$$DTIME(f(n)) \subset DTIME(g(n))$$

Proof: Same as before $\square$

Theorem 4

If f, g are time-constructible and $\lim_{n\to\infty} \frac{f(n+1)}{g(n)} = 0$, then

$$NTIME(f(n)) \subset NTIME(g(n))$$

Proof: Tricky because cannot just "flip" the output of a universal NDTM. Read proof $\square$

Theorem 5 (Ladner's theorem)

If $P \neq NP$, then there is $L \in NP \setminus P$ and L is not NP-complete.

Proof: On the board! $\square$

All diagonalization techniques must rely on the following two properties of TMs:

1. TMs are represented by strings
2. There is a universal TM that simulate any other without much running time/space overhead

Definition 6 (Oracle TM)

Oracle TM (or NDTM) for language O has an oracle tape where input $q \in \{0,1\}^*$ is written, and then the TM decides $q \in O$ in a single step.

Definition 7

For any language O:

- P^O = set of languages decided by polynomial TM with oracle access to O
- NP^O = set of languages decided by polynomial NDTM with oracle access to O

- $\overline{SAT} \in P^{SAT}$
- If $O \in P$, then $P^O = P$
- $EXPCON = \{\langle M, x, 1^n \rangle : M(x) = 1 \text{ in } 2^n \text{ steps}\}$

$P^{EXPCON} = EXP$: If $L \in EXP$
$\Rightarrow$ there is TM M that decides $x \in L$ in 2^{n^c} time
$\Rightarrow$ ask $EXPCON$ oracle question $\langle M, x, 1^{n^c} \rangle$
$\Rightarrow L \in P^{EXPCON}$
$\Rightarrow EXP \subseteq P^{EXPCON}$.

$NP^{EXPCON} = EXP$: If $L \in NP^{EXPCON}$
$\Rightarrow$ there is NDTM M^{EXPCON} that decides $x \in L$ in poly-time
$\Rightarrow$ can simulate both M and $EXPCON$ in EXP
$\Rightarrow NP^{EXPCON} \subseteq EXP$

$\Rightarrow P^{EXPCON} = NP^{EXPCON} = EXP$

Theorem 8

There exist oracles A, B, such that $P^A = NP^A$ and $P^B \neq NP^B$.

...i.e., result $P \stackrel{?}{=} NP$ cannot be extended to oracles (cannot be a relativizing result)

Proof: On the board! $\square$