**Def 7.1:** Let $M$ be a deterministic TM that always halts. The **running time** (or **time complexity**) of $M$ is the function $f: \mathbb{N} \rightarrow \mathbb{N}$, where $f(n)$ is the max number of steps $M$ takes on any input of length $n$.

**Note:** we usually use the big-$O$ notation, instead of precisely determining $f$

\[
\begin{align*}
\text{TIME}(n) & \quad \text{- problem can be solved in linear-time (O(n))} \\
\text{TIME}(n^2) & \quad \text{- quadratic (O(n^2))}
\end{align*}
\]

**Def 7.7:** The **time complexity class** $\text{TIME}(t(n))$ is the collection of languages that have an $O(t(n))$ deterministic decider (TM that always halts).
Measuring Complexity

What about nondeterministic TMs?
Measuring Complexity

What about nondeterministic TMs?

**Def 7.9:** Let $N$ be a nondeterministic decider. The *running time* of $N$ is the function $f:N \rightarrow N$, where $f(n)$ is the maximum number of steps that $N$ uses on any branch of its computation on any input of length $n$.

**Thm 7.11:** Let $t(n)$ be a function, where $t(n) \geq n$. Then every $t(n)$ nondeterministic single-tape TM has an equivalent exponential-time deterministic single-tape TM.