Decision vs Optimization

In algorithms, we are usually concerned with optimization.
- Shortest path
- Minimum spanning tree
- Maximum flow

There is a corresponding decision problem variant.
- Does $G$ have a path of length at most $k$?
- Does $G$ have a MST of weight at most $k$?
- Does $G$ have a flow of at least $k$?

Decision problems can be simple yes/no questions.
- SAT
- Hamilton path
- Halting language

Formal Languages

An **alphabet** $\Sigma$ is a finite set of symbols.

A **string** over an alphabet $\Sigma$ is a finite sequence of symbols in $\Sigma$.

**Example** $\Sigma = \{0, 1\}$, $011011000000$

The set of all strings over an alphabet $\Sigma$ is $\Sigma^*$.

**Example** $\Sigma^* = \{\lambda, 0, 1, 00, 01, 10, 11, 000, \ldots\}$

The empty string is $\epsilon$ (or $\lambda$).

The set of all strings over $\Sigma$ except $\epsilon$ is $\Sigma^+$.

A language over $\Sigma$ is a subset of $\Sigma^*$.
We can encode decision problems as languages. A yes/no decision corresponds to in/out of the language.

Ex: Given $x, y, z$, does $x + y = z$?

$$\Sigma = \mathbb{Z}_0 \cup \{0, 1\}$$

$$\text{ADD} = \{x + y = z \mid x, y, z \in \mathbb{Z}_0^+ \text{ and } x + y = z\}.$$

Usually, the alphabet we use is $\{0, 1\}$. Any alphabet we might choose can be encoded into binary strings with only a constant factor slowdown.