- Reductions

- A can be reduced to B, if given a solution to B, can use that to solve problem A.

```plaintext
B (input to prob. B)
|
|
|
return correct answer given to you
```

```plaintext
A (input to problem A)
|
|
|
do anything on input
|
call B( ... )
|
do something with output
|
call B(1, 2)
|
return correct answer to A
```

- Example: LIS to LCS

\[ X = \{ 5, 2, 3, 6, 4, 9 \} \]

\[ \text{LIS} = 4, \{2, 3, 6, 9\} \]

- Reduction

\[ \text{LIS}(X[1]) \]

| \{ \}
| \{ \}
| \{ \}

\[ Y = \{ 2, 3, 4, 5, 6, 9 \} \]

\[ \text{LCS}(\{5, 2, 3, 6, 4, 9\}, \{2, 3, 4, 5, 6, 9\}) = 4, \{2, 3, 6, 9\} \]

\[ \text{(best) runtime for LIS} \leq \text{(best) runtime for LCS + } O(n^2) \]

- A can be reduced to B, reduction time "small" if A is easier than B "easier" "no harder than"
runtime $A \leq 0$ (runtime for $B$)

- complexity class, easy vs. hard problems
  - P: set of decision problems that can be solved in polynomial time.
  - NP: set of decision problems whose solution can be verified in polynomial time.

  Accept if solution is correct.

  \[
  \begin{array}{c}
  \text{ verifier } \\
  \text{ polynomial time }
  \end{array}
  \quad \Rightarrow \quad \begin{array}{c}
  \text{ solution } \\
  \text{ output of NP problem }
  \end{array}
  \]

  YES if $\exists$ solution s.t. verifier(input, solution) accepts
  NO if for any solution verifier(input, solution) rejects.

- $P \subseteq \text{NP}$, believe $P \neq \text{NP}$

- Polynomial time reduction: convert input $X$ of $A$ to input $Y$ of $B$ in poly time, return $B(Y)$. 