Lecture 21 Complexity, Reductions

Monday, April 13, 2020 3:01 PM

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

```
B(input to prob. B)
{
   . . . .
   . . . .
   return correct answer to B
}
```

given to you

```
A(input to problem A)
{
   do anything on input
   call B( ... )
   do something with output
   call B(...)
   . . . .
   return correct answer to A
}
```

- example: LIS to LCS

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

LIS = 4 $\{2, 3, 6, 9\}$

- reduction

```
LIS(X[])
{
   Y = MergeSort(X)
   return LCS(X, Y)
}
```

```
LCS(X[], Y[])
{
   . . .
}
```

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

$LCS(\{5, 2, 3, 6, 4, 9\}, \{2, 3, 4, 5, 6, 9\})$

$= 4 \quad \{2, 3, 6, 9\}$

n^2

(best) runtime for LIS ≤ (best) runtime for LCS + $\Theta(n \log n)$

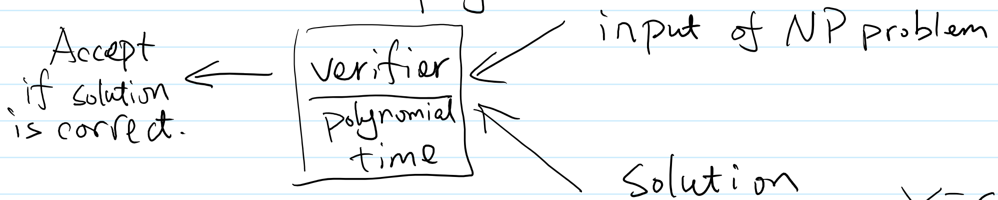
- A can be reduced to B, reduction time "small"

A is "easier" than B

"easier" "no harder than" $\lesssim$

runtime $A \lesssim O(\text{runtime for } B)$

- 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.



output of NP problem
- YES: if ∃ solution s.t. verifier(input, solution) accepts
- NO: if for any solution verifier(input, solution) rejects.

- $P \subseteq NP$, believe $P \subset NP$

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