Lecture 6: Enumerative Synthesis with Pruning

Yu Feng

Spring 2020

Summary of previous lecture

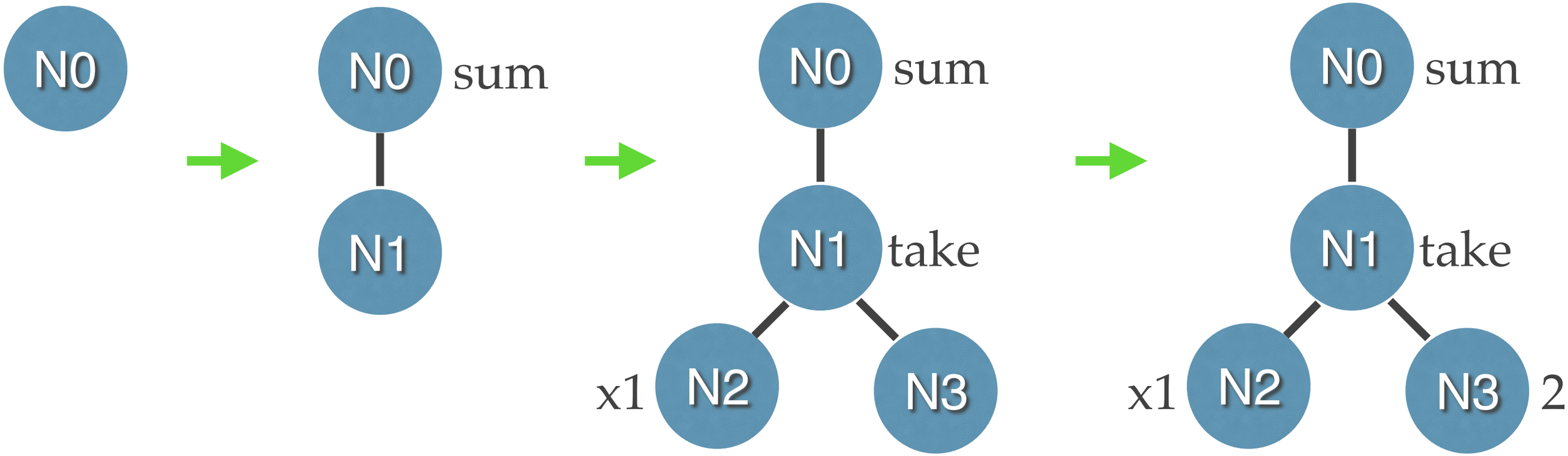
- Inductive program synthesis
  - Enumerative search
  - Symbolic search
  - Neural-guide search

For all inputs x, find a program P that meets the specification ϕ

$$\exists\ P.\ \forall\ x.\ \phi(x, P(x))$$

$$
\begin{aligned}
N \rightarrow\ & \texttt{0} \mid \ldots \mid \texttt{10} \mid x_i \mid \texttt{last}(L) \mid \texttt{head}(L) \mid \texttt{sum}(L) \\
& \mid \texttt{maximum}(L) \mid \texttt{minimum}(L) \\
L \rightarrow\ & \texttt{take}(L, N) \mid \texttt{filter}(L, T) \mid \texttt{sort}(L) \mid \texttt{reverse}(L) \mid x_i \\
T \rightarrow\ & \texttt{geqz} \mid \texttt{leqz} \mid \texttt{eqz}
\end{aligned}
$$

$$
\begin{aligned}
N \rightarrow\ & \texttt{0} \mid \texttt{...} \mid \texttt{10} \mid x_i \mid \texttt{last}(L) \mid \texttt{head}(L) \mid \texttt{sum}(L) \\
& \mid \texttt{maximum}(L) \mid \texttt{minimum}(L) \\
L \rightarrow\ & \texttt{take}(L, N) \mid \texttt{filter}(L, T) \mid \texttt{sort}(L) \mid \texttt{reverse}(L) \mid x_i \\
T \rightarrow\ & \texttt{geqz} \mid \texttt{leqz} \mid \texttt{eqz}
\end{aligned}
$$



Top-down enumerative synthesis

```
Synthesize(inputs, outputs){
 wlist := start_symbol
 while(true):
  Deque p from wlist;
  if(isConcrete(p))
   if(isCorrect(p, inputs, outputs))
    return p;
  else
   wlist := wlist U grow(p);
}
```

Which one we should pick?

How to grow?

For all inputs x, find a program P that meets the specification ϕ

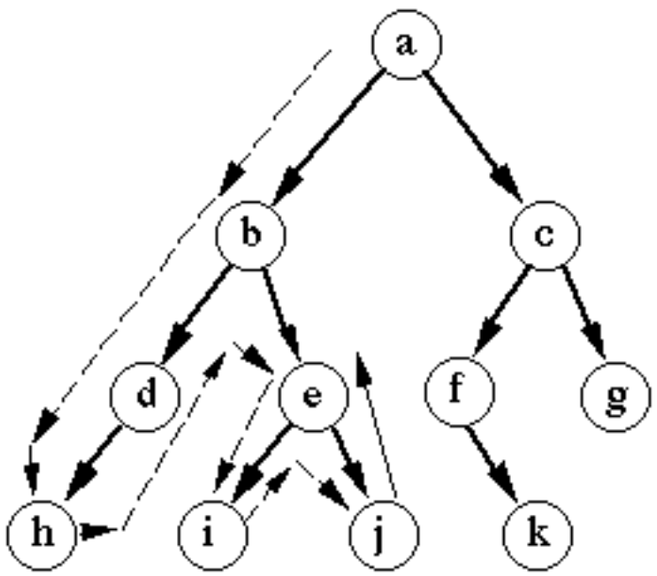
$$\exists\, \mathrm{P}.\ \forall\ \mathrm{x}.\ \phi(\mathrm{x},\ \mathrm{P}(\mathrm{x}))$$

$$
\begin{aligned}
N \rightarrow\ & \texttt{0} \mid \ldots \mid \texttt{10} \mid x_i \mid \texttt{last}(L) \mid \texttt{head}(L) \mid \texttt{sum}(L) \\
& \mid \texttt{maximum}(L) \mid \texttt{minimum}(L) \\
L \rightarrow\ & \texttt{take}(L, N) \mid \texttt{filter}(L, T) \mid \texttt{sort}(L) \mid \texttt{reverse}(L) \mid x_i \\
T \rightarrow\ & \texttt{geqz} \mid \texttt{leqz} \mid \texttt{eqz}
\end{aligned}
$$

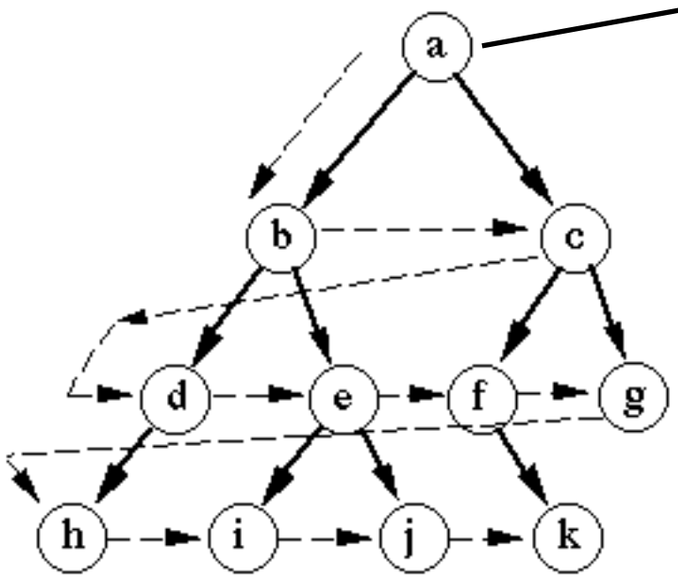
$$2^{50}$$

Better search orders

Enumerative search on AST



DFS



BFS

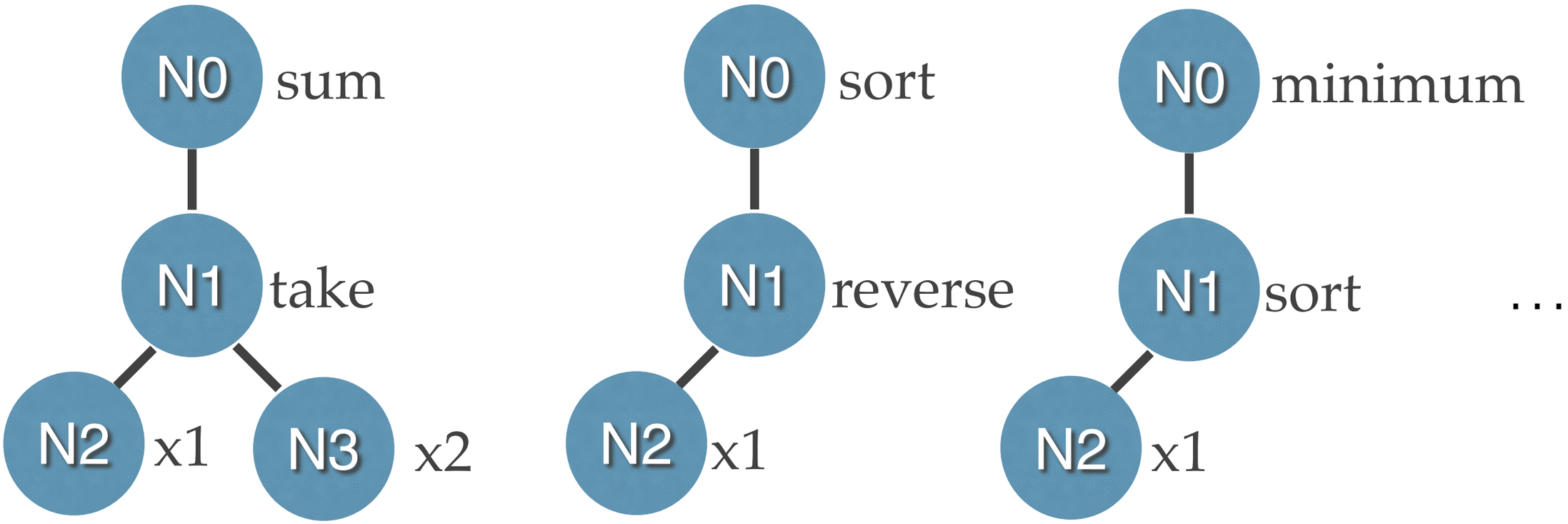
sum or head?

N	$\ldots$	10	`last`(L)	`head`(L)	`sum`(L)
`maximum`(L)	`minimum`(L)				

1. How to guide enumerative search?

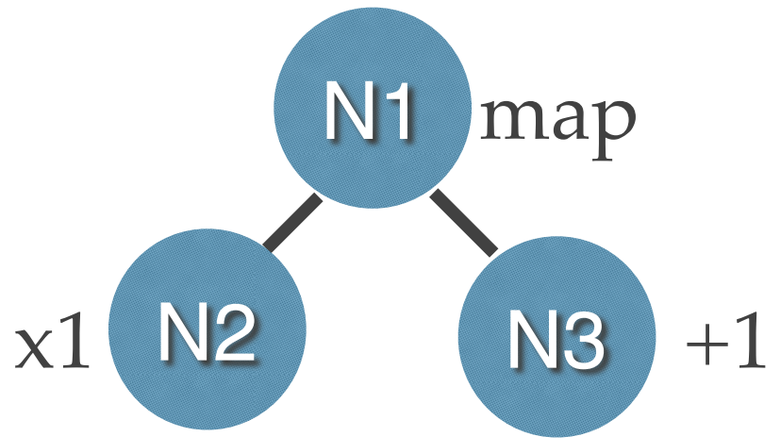
Better pruning

How to cut search space?



Better pruning

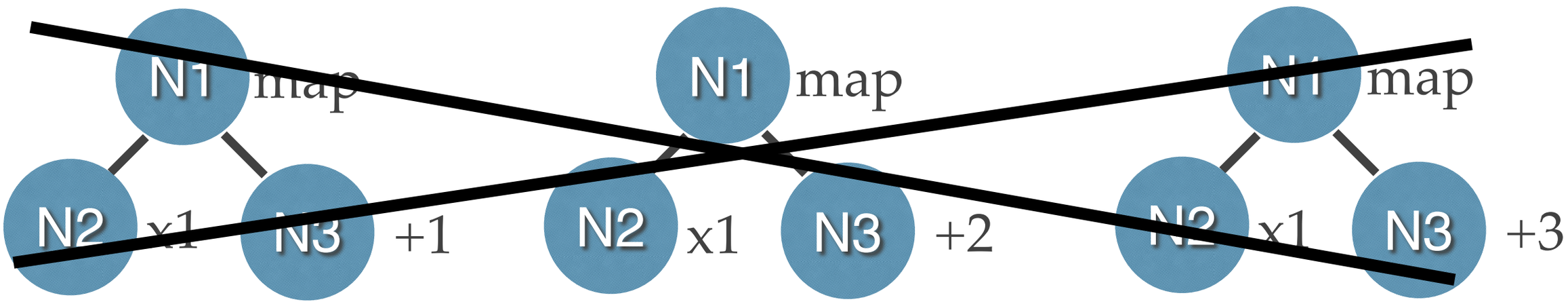
How to cut search space?



map([1,2,3], +1) = [2,3,4]

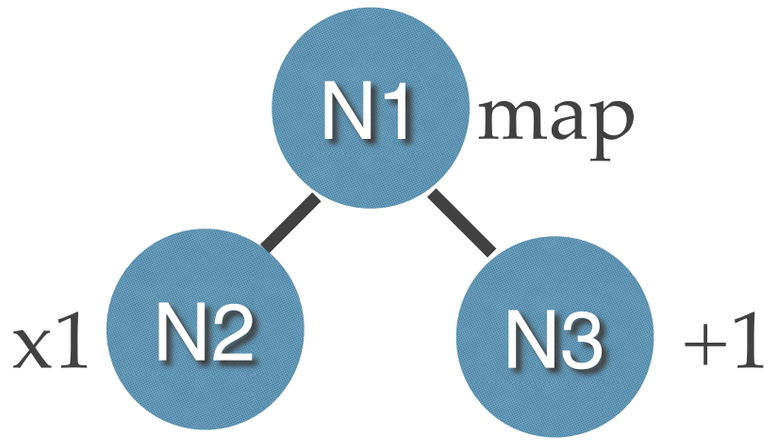
map([2,3,4], +2) = [4,5,6]

$$f([1,2,3]) = [4,5]$$



Better pruning

How to cut search space?

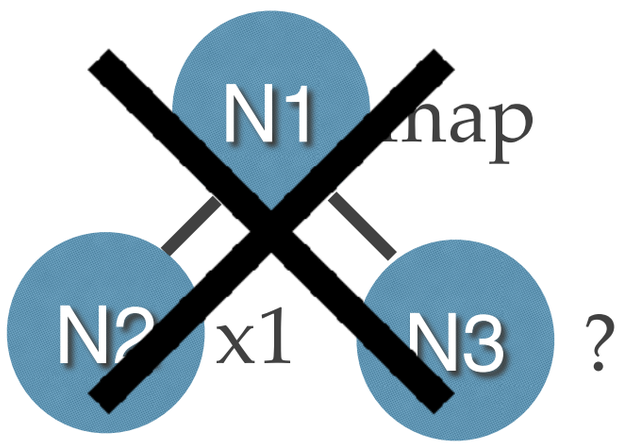


map([1,2,3], +1) = [2,3,4]

map([2,3,4], +1) = [3,4,5]

Apply map will preserve the size of the input!

2. How to prune partial program?



Synthesis with pruning

```
Synthesize(inputs, outputs){
 wlist := start_symbol
 while(true):
  Deque p from wlist;
  if(isConcrete(p))
   if(isCorrect(p, inputs, outputs))
    return p;
  else
   if(isValid(p))wlist := wlist U grow(p);
}
```

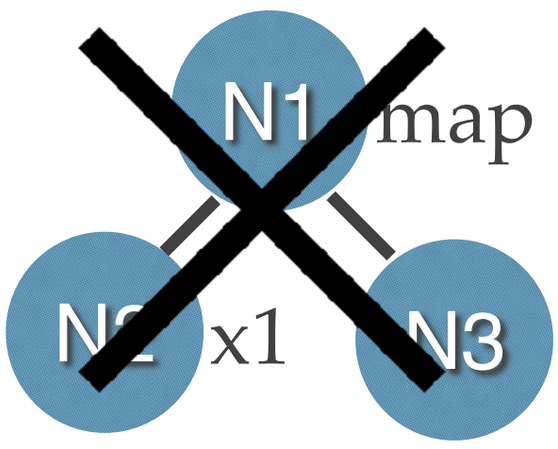
Prune infeasible partial programs!

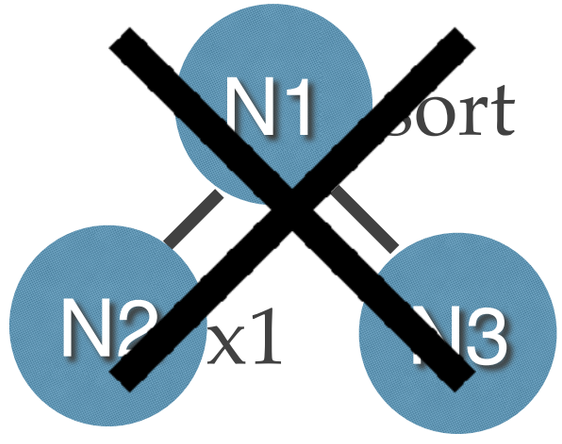
Learning from past mistake

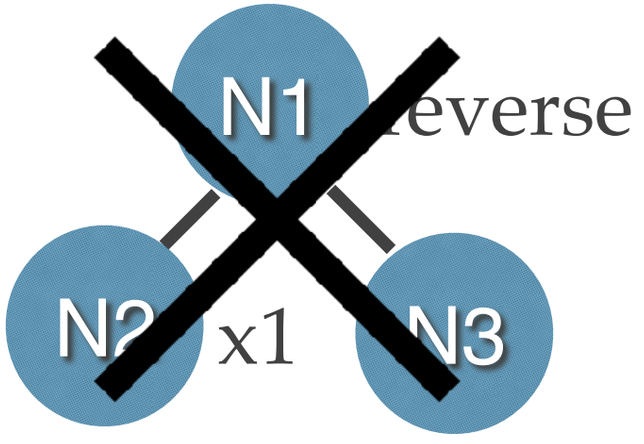
Can we further prune the search space?

$$f([1,2,3]) = [4,5]$$

3. How to learn from past mistakes?







TODOs by next lecture

- HW2 will be out
- HW1 will be due
- Discuss your final project during office hour