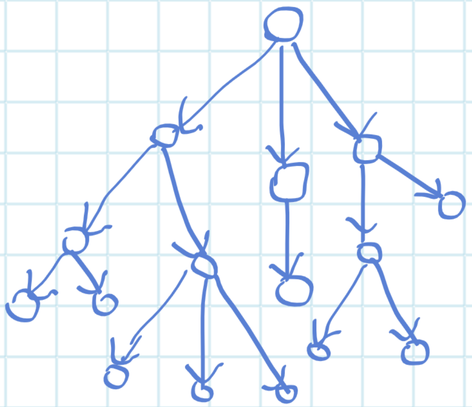
# Directed Tree Graph

## DTG

### Structure

node

Parent  Child[n]



Each node is a class

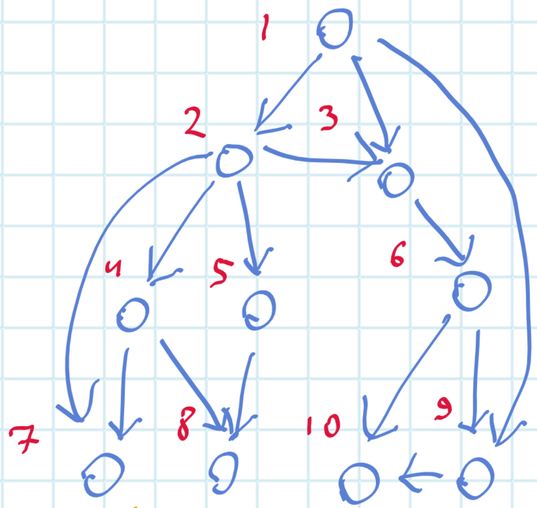
# Directed Acyclic Graph

## DAG

### Structure

node

Parent[n]




Each node is a class

# Topological DAG

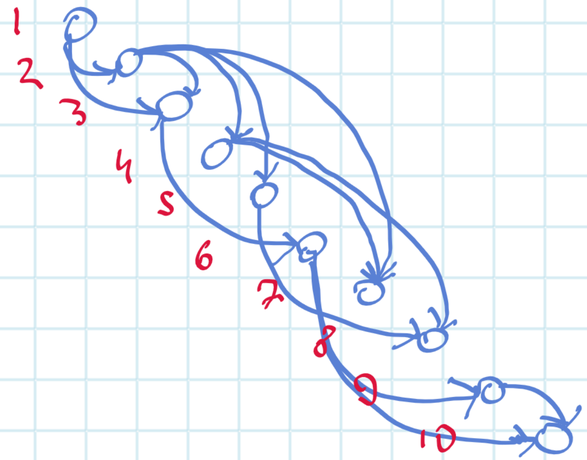
## Data Oriented Design-

## DOD friendly

### Structure

node

Parent[n]




Each node can be an integer reference to a type

[1, 9, 3, 4, 2, 7, 10, 5, 6, 8,
5', 7, 2', 3', 9', 1', 4', 8', 6']

1 → [9, 3, 2] → 2

2 → [9, 3, 4, 7, 5] → 4

4 → [5, 9, 3, 7, 8] → 5

5 → [7, 9, 3, 8] → 7

7 → [8, 9, 3] → 8

8 → [9, 3] = 9 {6, 7'}

9 6 = 6 {3}

6 3 = 3 {3'}

3 3' = 3' {2' 1'}

3' 2' = 2' {1'}

2' 1' = 1' → 1'

1' → [2', 3', 4'] → 2'

2' → [4', 3'] → 4'

4' → [3', 5', 7', 6'] → 3'

3' → [5', 7', 6', 3] → 5'

5' → [3, 7', 6', 9', 8'] → 3

3 → [7', 8', 6', 9', 6] → 7'

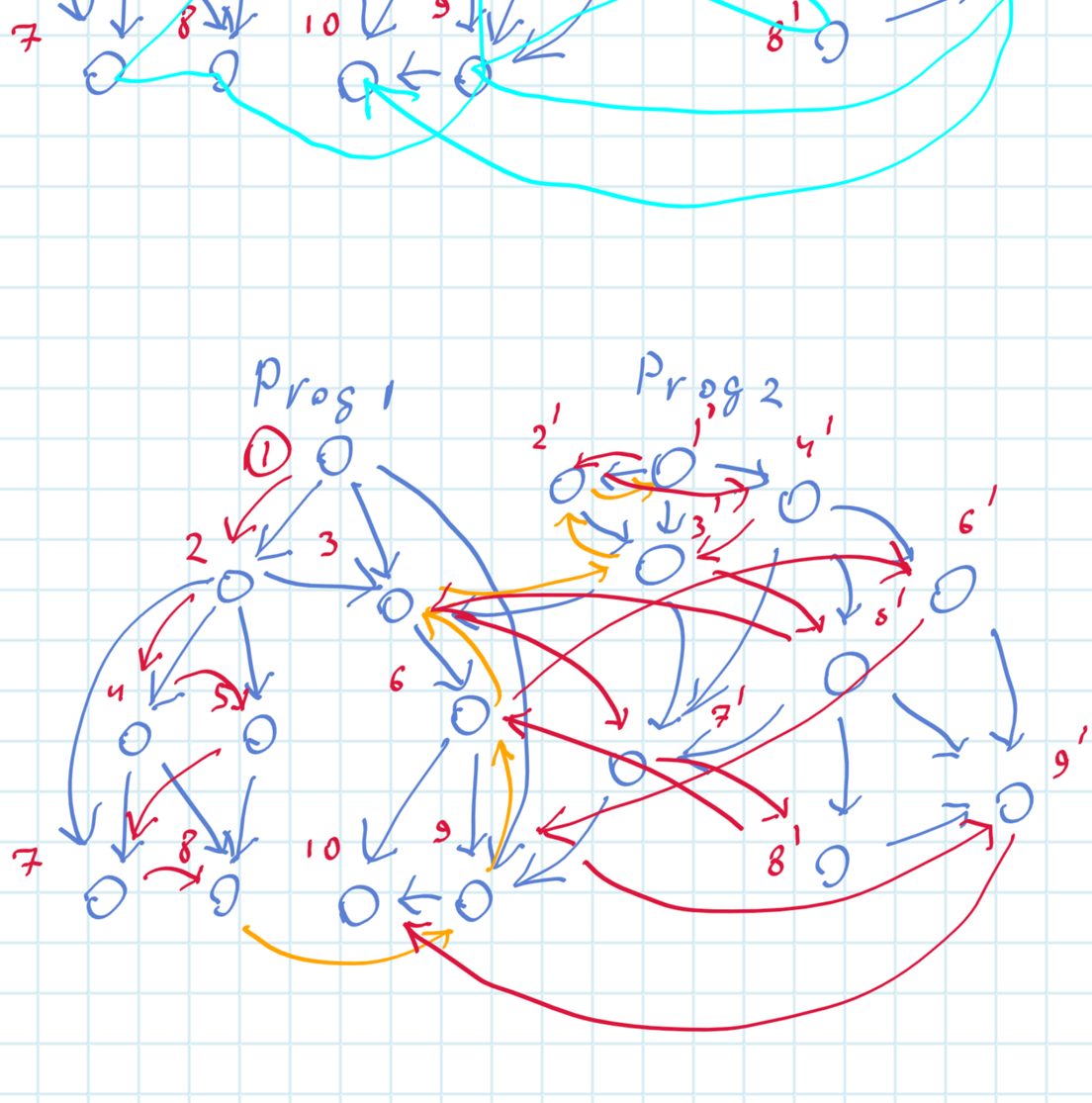
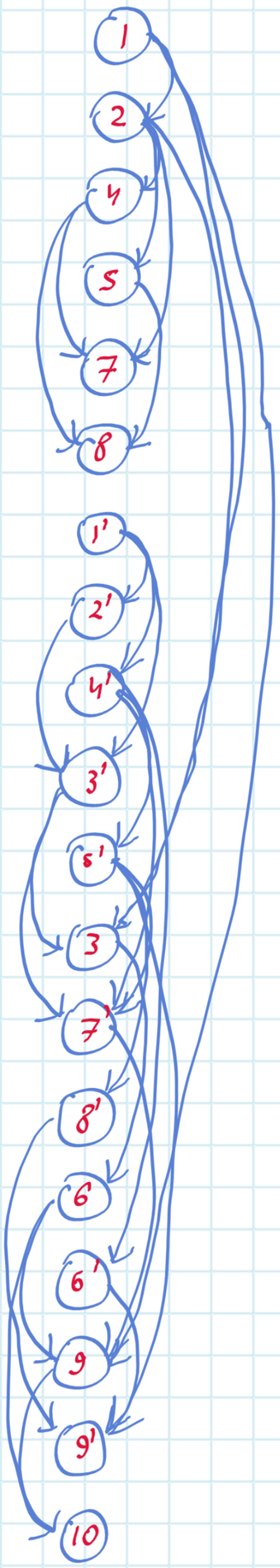
7' → [8', 6, 6', 9', 9] → 8'

8' → [6, 6', 9', 9] → 6

6 → [6', 9', 9, 10] → 6'

6' → [9, 9', 10] → 9

9 → [9', 10] → 9'

9' → [10] → 10

10 EOF