## Acorn - seed:list[int] - state:dict - M:int - seen:set + Acorn(seed:int, M:int) + get state():dict + set state(state:dict) Analyzer + Analyzer(rand\_num\_gen:(MiddleSquare || LaggedFibonacci || Acorn || LinearCongruential )

## LaggedFibonacci

- seed:list[int]
- state:dict
- seen:set
- operator:str
- j:int
- k:int

\_\_\_\_

- m:int
- val:list[int]
- + LageedFibonacci(seed:list[int], j:int, k:int, m:int)
- + get state():dict
- + set\_state(state:dict)

## LinearCongruential

- seed:int
- state:dict
- seen:set
- a:int

- max:int

- min:int

- period:int

- average:float

- bit\_freqs:list[int]

+ analyze(max\_nums:int)

MiddleSquare

+ MiddleSquare(seed:int)

+ set\_state(state:dict)

+ get\_state():dict

- seed:int

- seen:set

- state:dict

- c:int
- m:int
- + LinearCongruential(seed:int, a:int, c:int, m:int)
- + get\_state():dict
- + set\_state(state:dict)