- Map N ⇒ N
- Filter N ⇒ ≤ N

* Reduce

- from functools import reduce

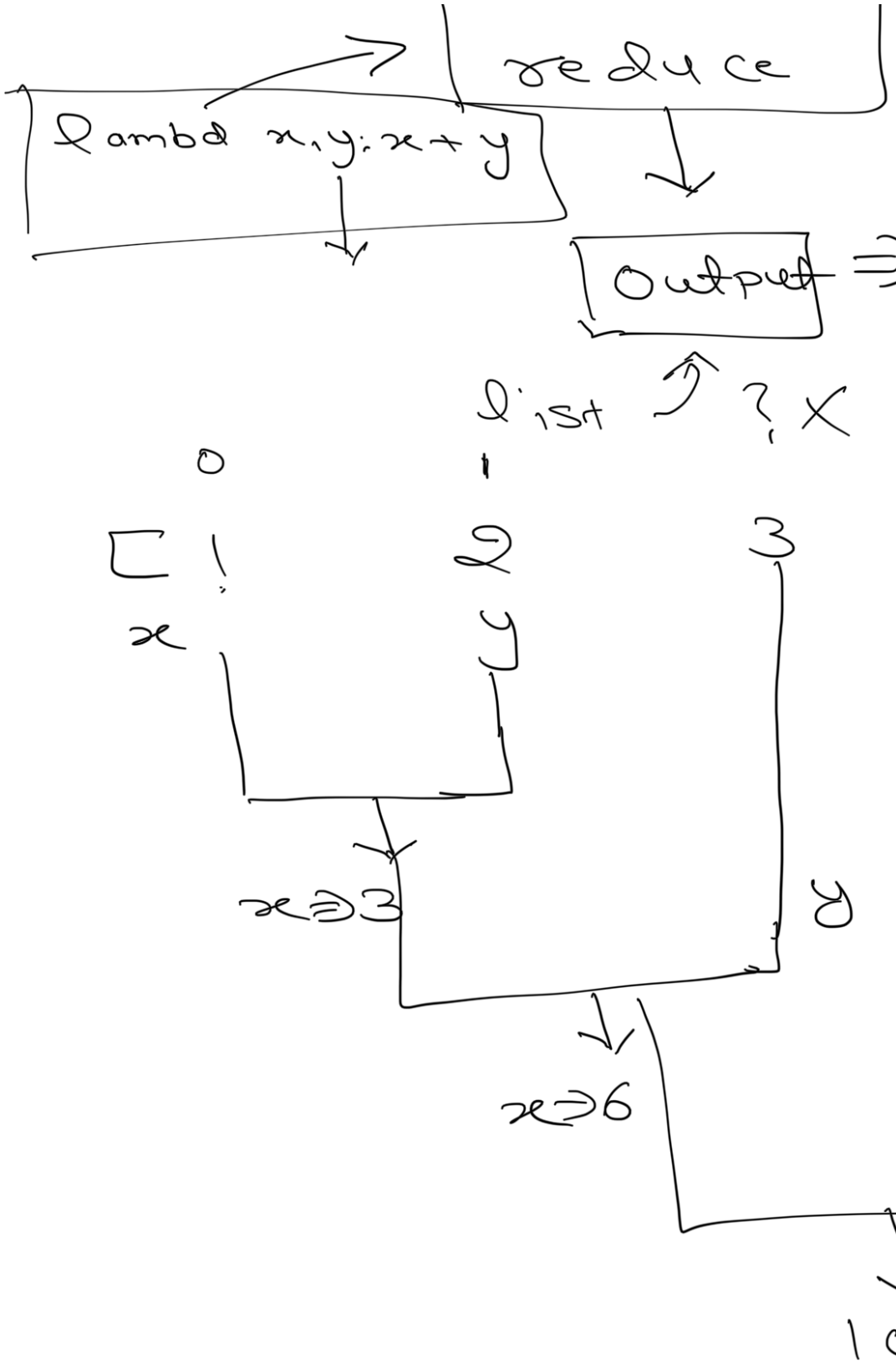
import functools.reduce as reduce

- reduce(funct, iterable)

takes 2 args

- reduce(lambda x, y : x + y, [1,2,3,4]

[1 2 3 4


reduce
lambda x,y: x+y
Output ⇒ 10
list ? x
0
1
x
y
x⇒3
y
x⇒6
y
10

reduce(lambda x,y:x+y, list)

⇒ 'sachin'

⇒ $\frac{n \times (n+1)}{2}$ ⇒ $\frac{10^5 (10+1)}{2}$

⇒ 55

Higher Order Function:

⇒ A function which return another Function

```
def gen_exp(n):
    def exp(x):
        return x**n
    return exp
```

(which takes x and raises to POWS

exp5 ⇒ gen_exp(5)

Func ⇒ exp_5(x=10) ⇒ 10^5

exp_5(2) ⇒ 2^5

exp_10 ⇒ gen_exp(10)

x^10

def gen_exp(5)

def exp(x):

ret x**5

ret exp

gen_exp(5)

a(2) ⇒ 2**5

b ⇒ gen-exp(10)

b ⇒ def exp(x):
    x**10

b(2) ⇒ 2**10