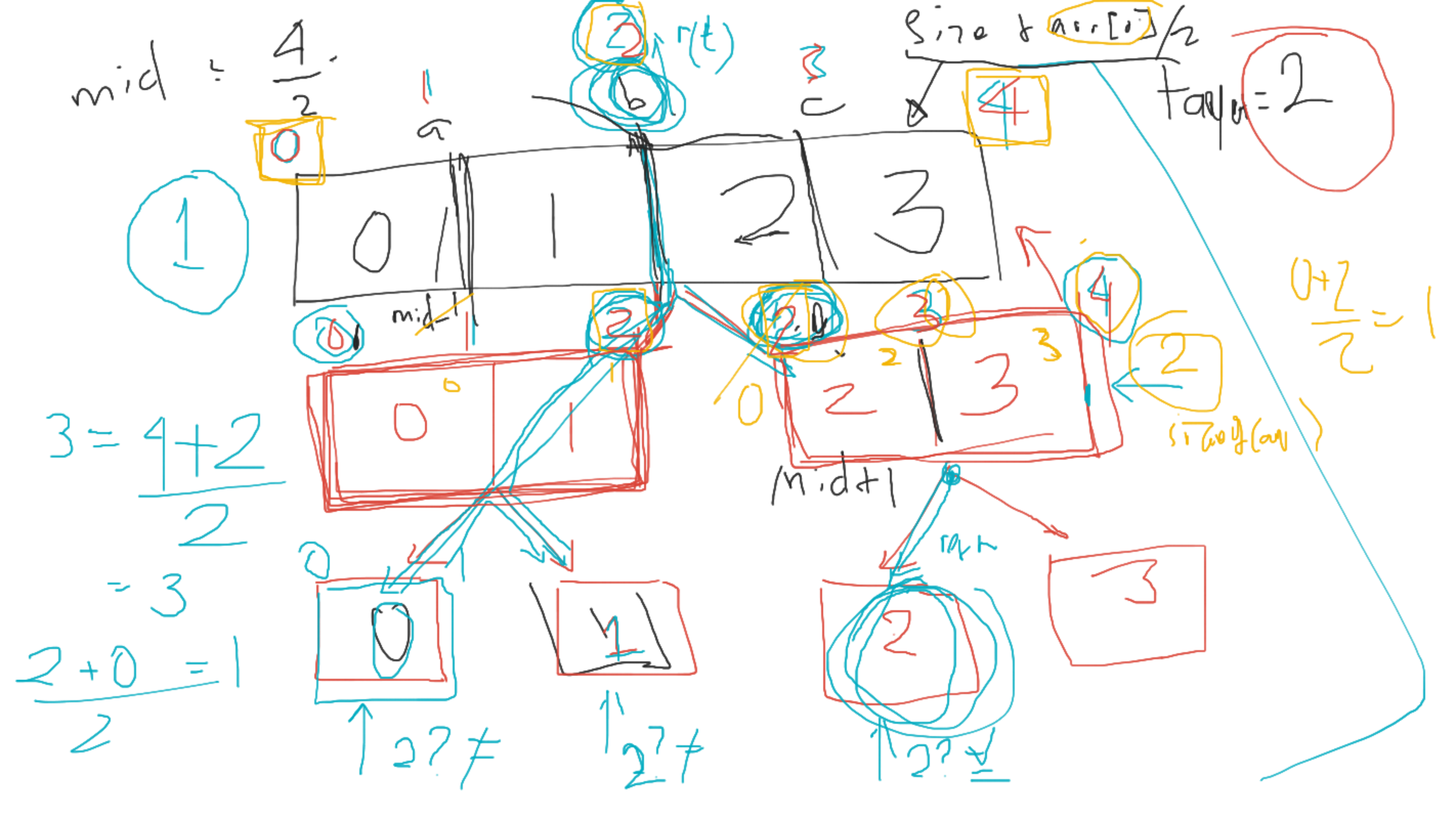
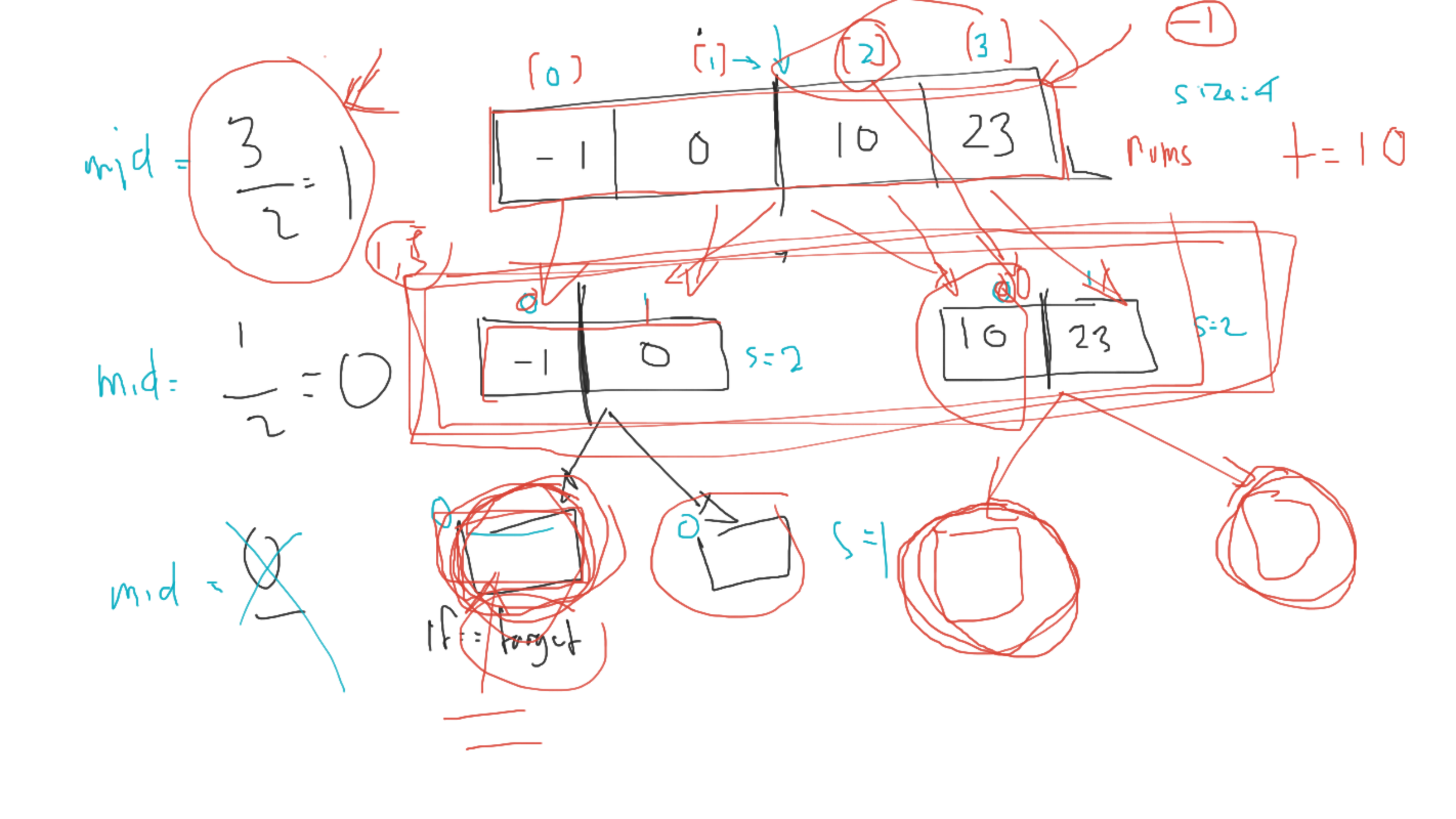
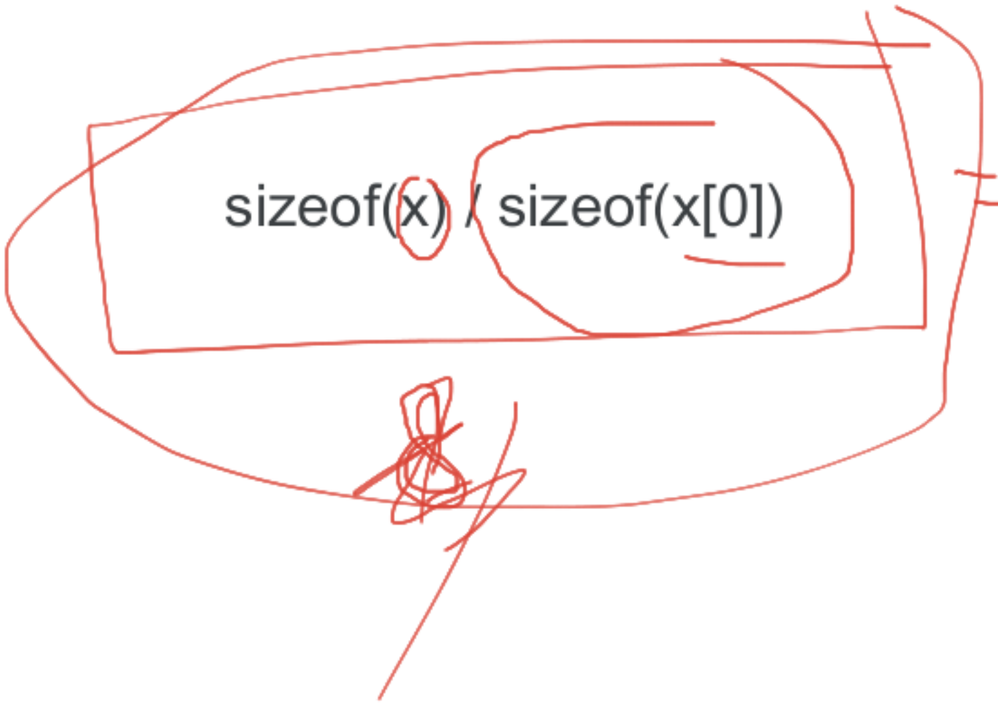
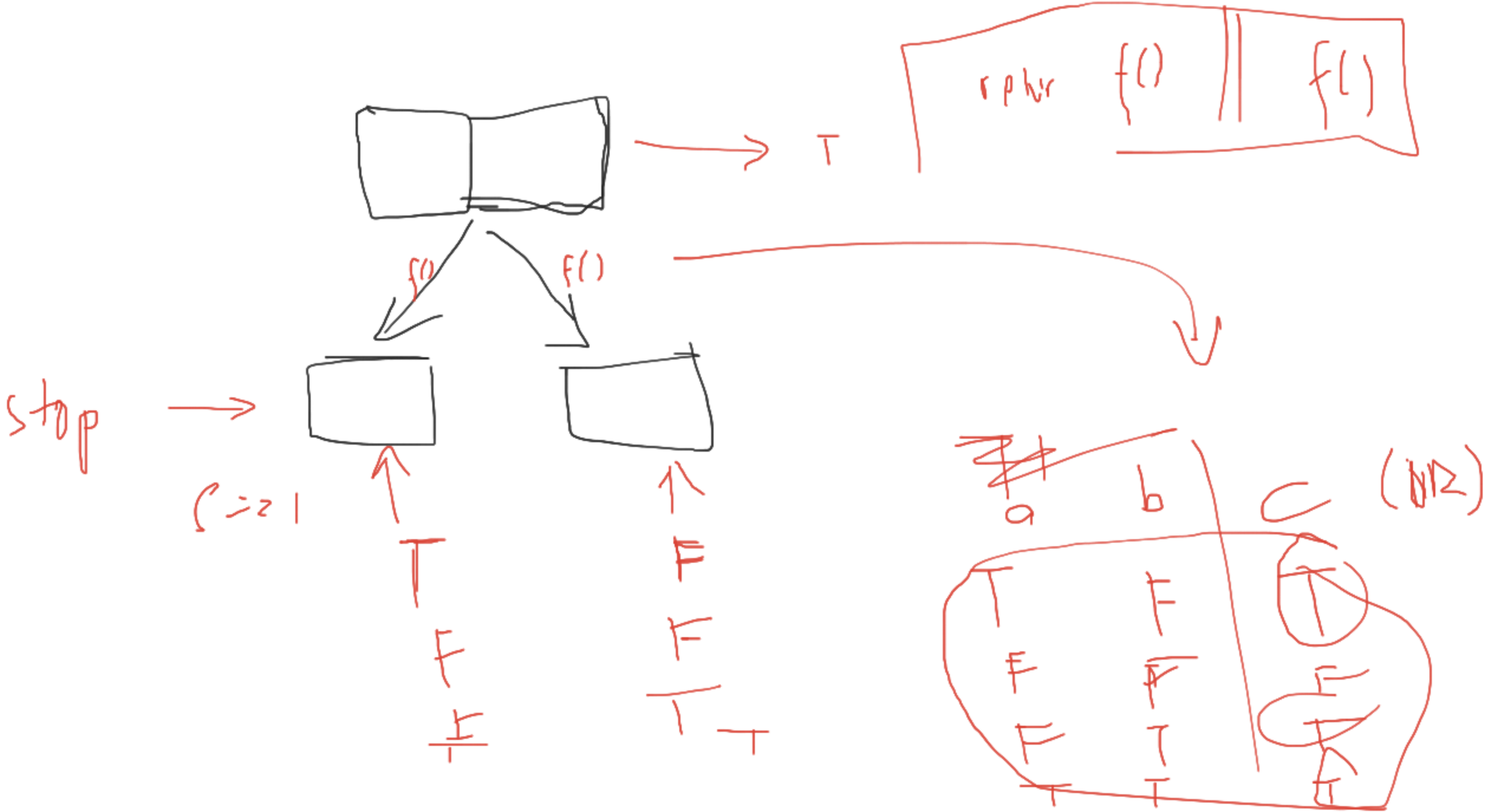
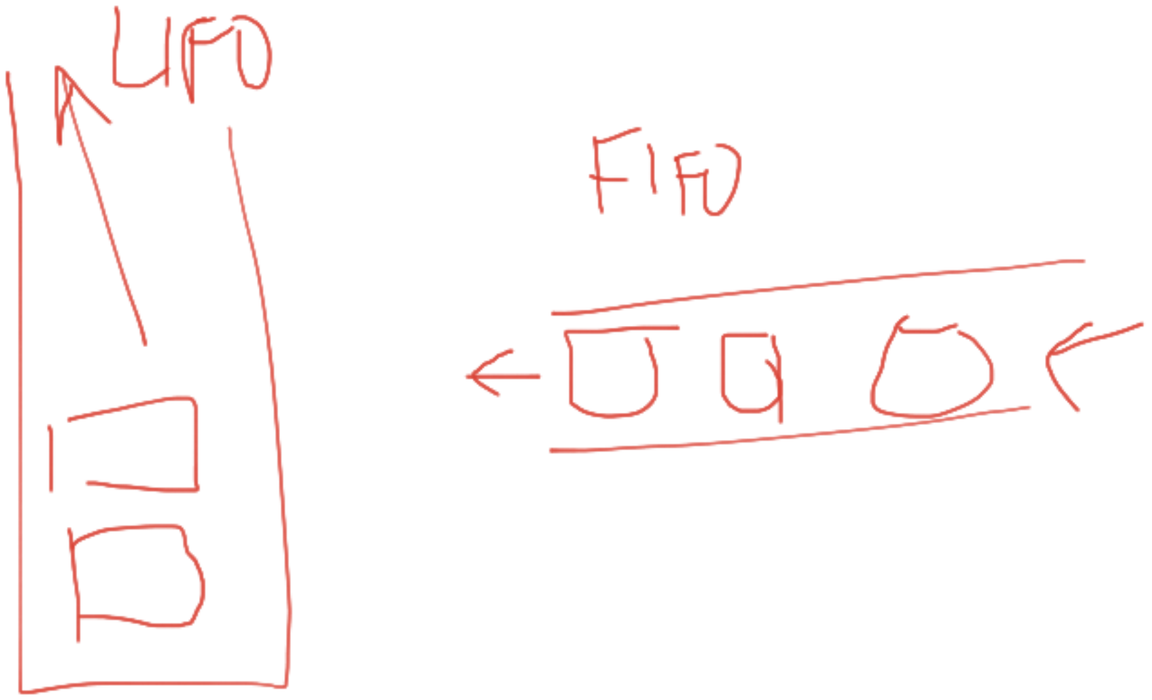

mid = 4/2
size of arr[0]/2
target = 2
r(t)
a
b
c
0
1
2
3
4
mid-1
mid+1
0+2/2 = 1
3 = (4+2)/2
= 3
(2+0)/2 = 1
2?≠
2?≠
2?≠

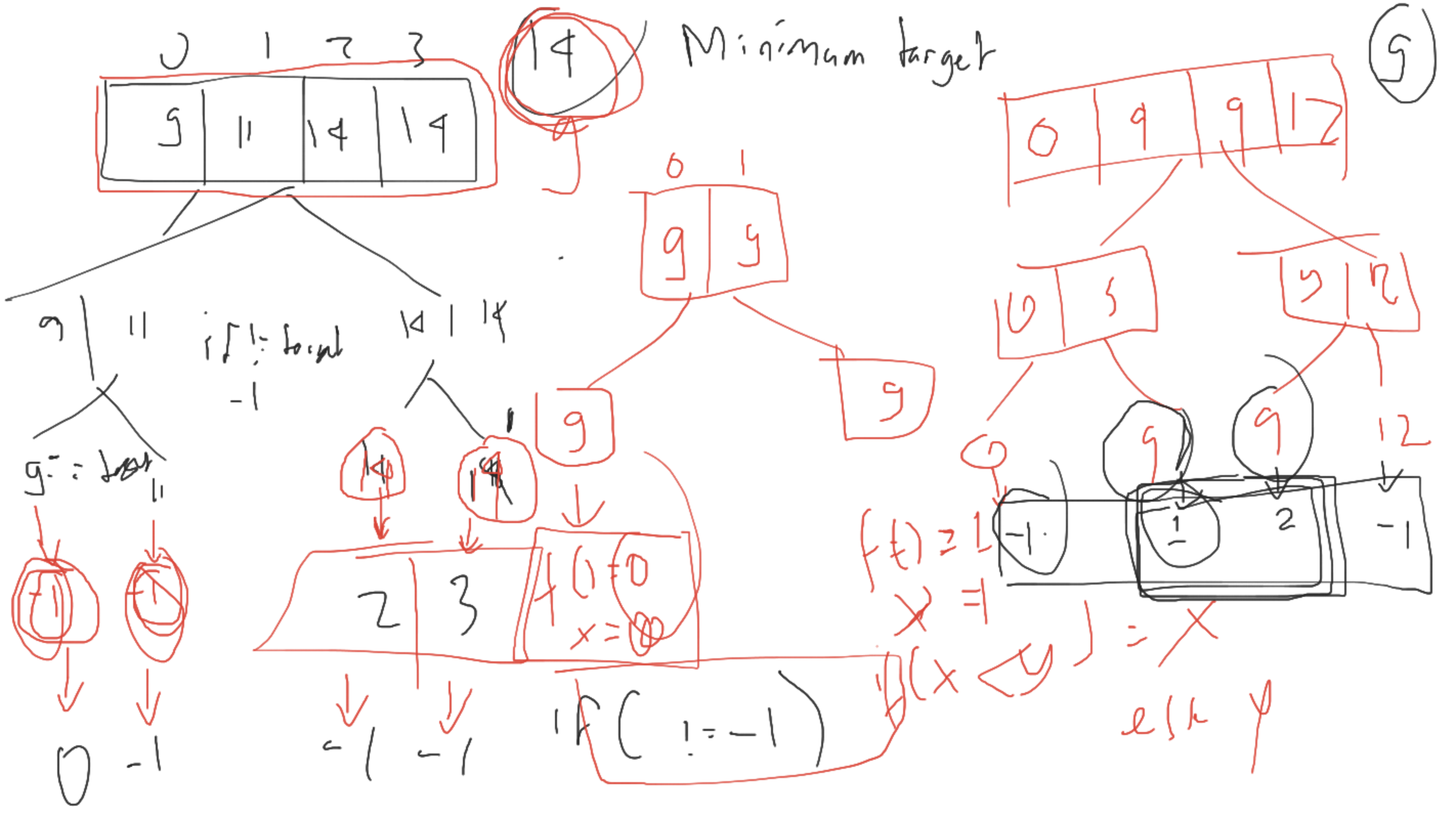

mid = 3/2 = 1
[0]
[1]
[2]
[3]
-1
size: 4
-1
0
10
23
nums
t = 10
mid = 1/2 = 0
s=2
10
23
s=2
mid =
s=1
if = target


sizeof(x) / sizeof(x[0])

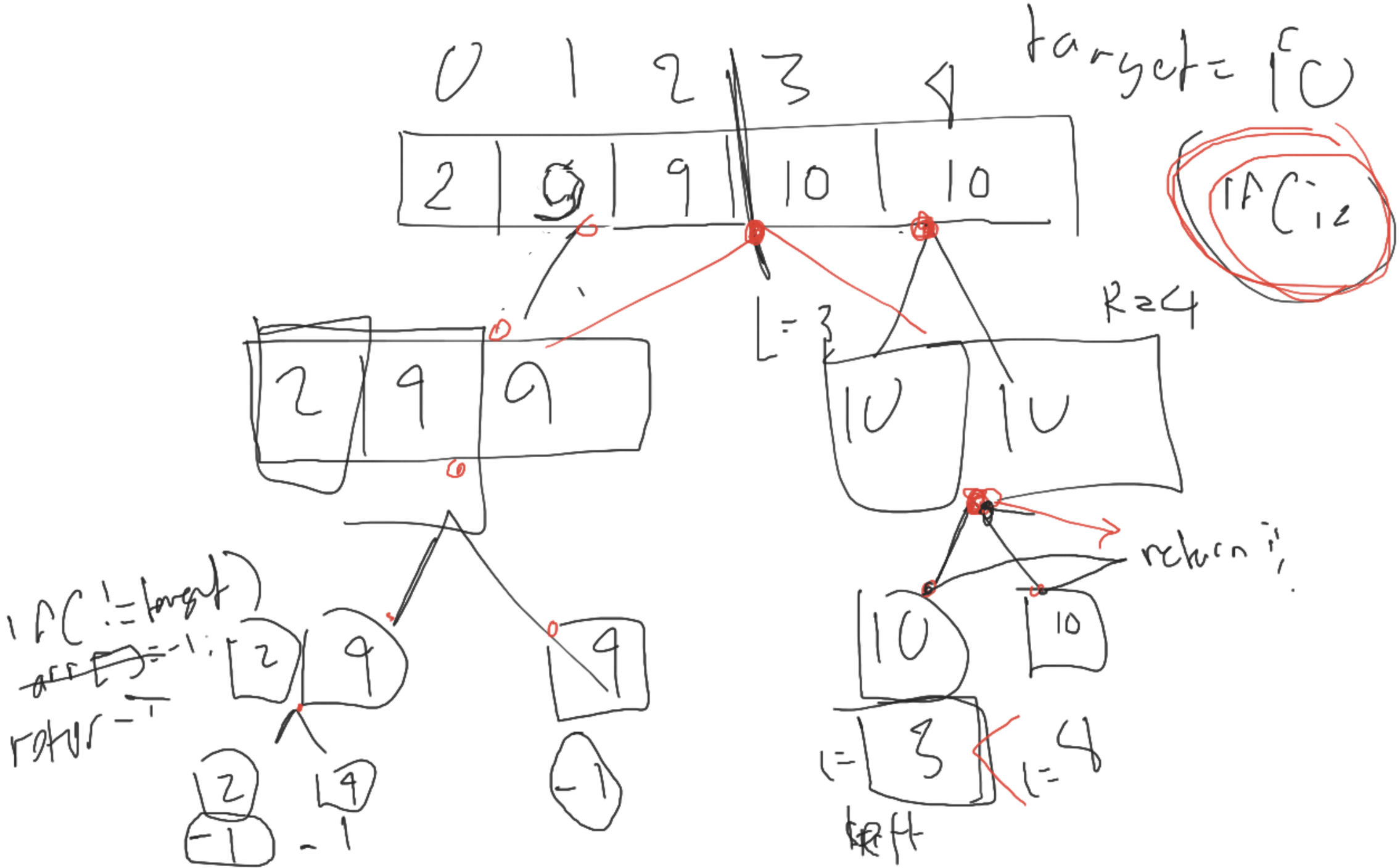


T
rphr f() || f()
f()
f()
stop
c := 1
T
F
F
F
F
T
T
a
b
c
(DR)
T F T
F F F
F T T
T T T

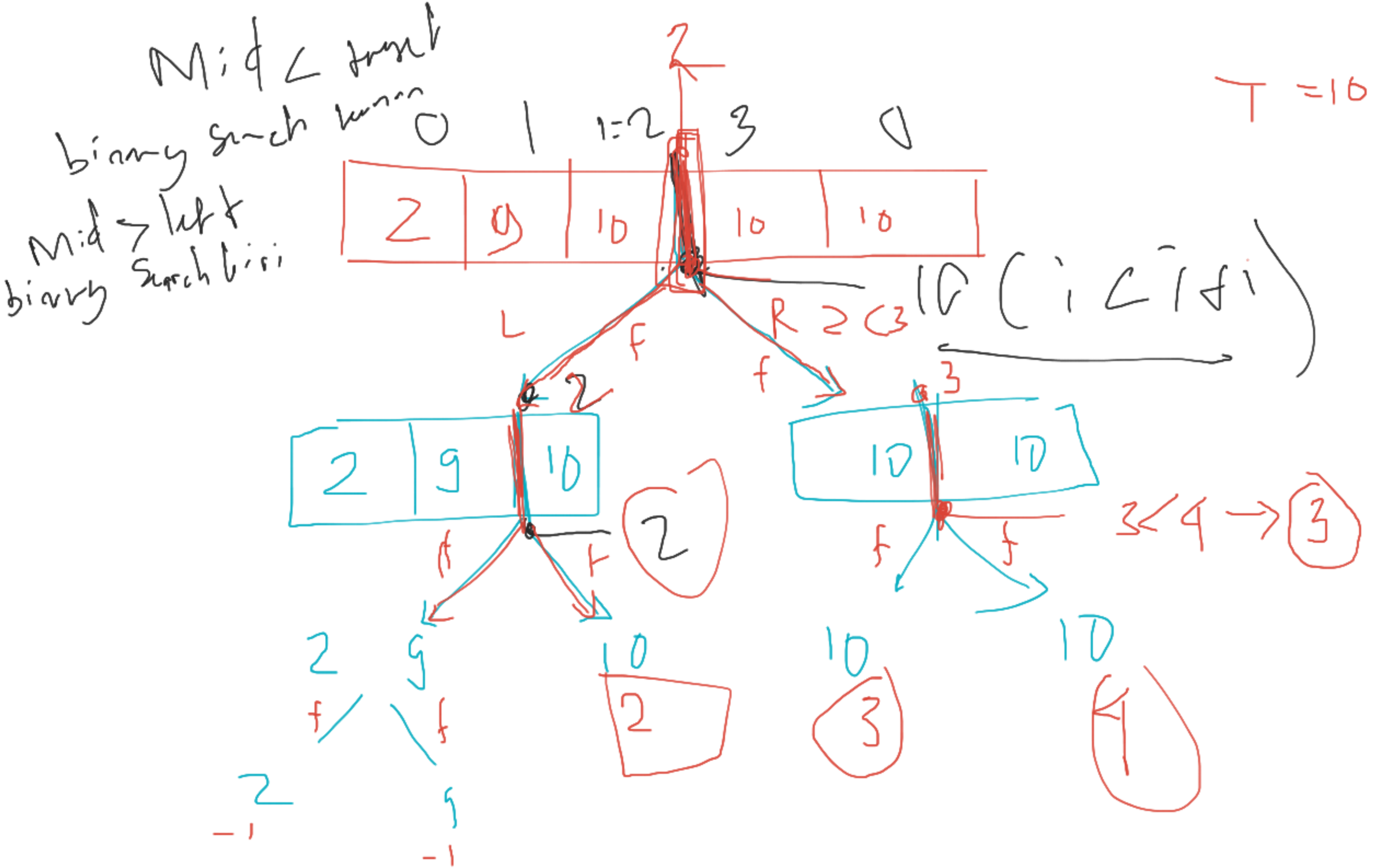

LIFO
FIFO



Minimum target
14
9
0 1 2 3
9 11 14 14
9 11
if != target
-1
14 14
9 = target
11
-1
-1
0 -1
14
14
2 3
-1 -1
f() = 0
x = 0
if(!= -1)
0 1
9 9
9
9
0 9 9 12
0 9
9 12
0
9
9
12
-1
1
2
-1
f() = 1
x = 1
if(x < y) = x
else y



target = 10
0 1 2 3 4
2 9 9 10 10
L = 3
R = 4
2 9 9
10 10
return i;
10
10
3
l = 4
left
2 9
9
2
4
-1
-1
-1


Mid < target
binary search
Mid > left
binary search
T = 10
0 1 1=2 3 0
2 9 10 10 10
L
f
R 2<3
f
2
2 9 10
3
10 10
f
f
f
f
2
3<4 → 3
2 9 10 10 10
f f
2 9
-1 -1
2
3
4

f(x) {

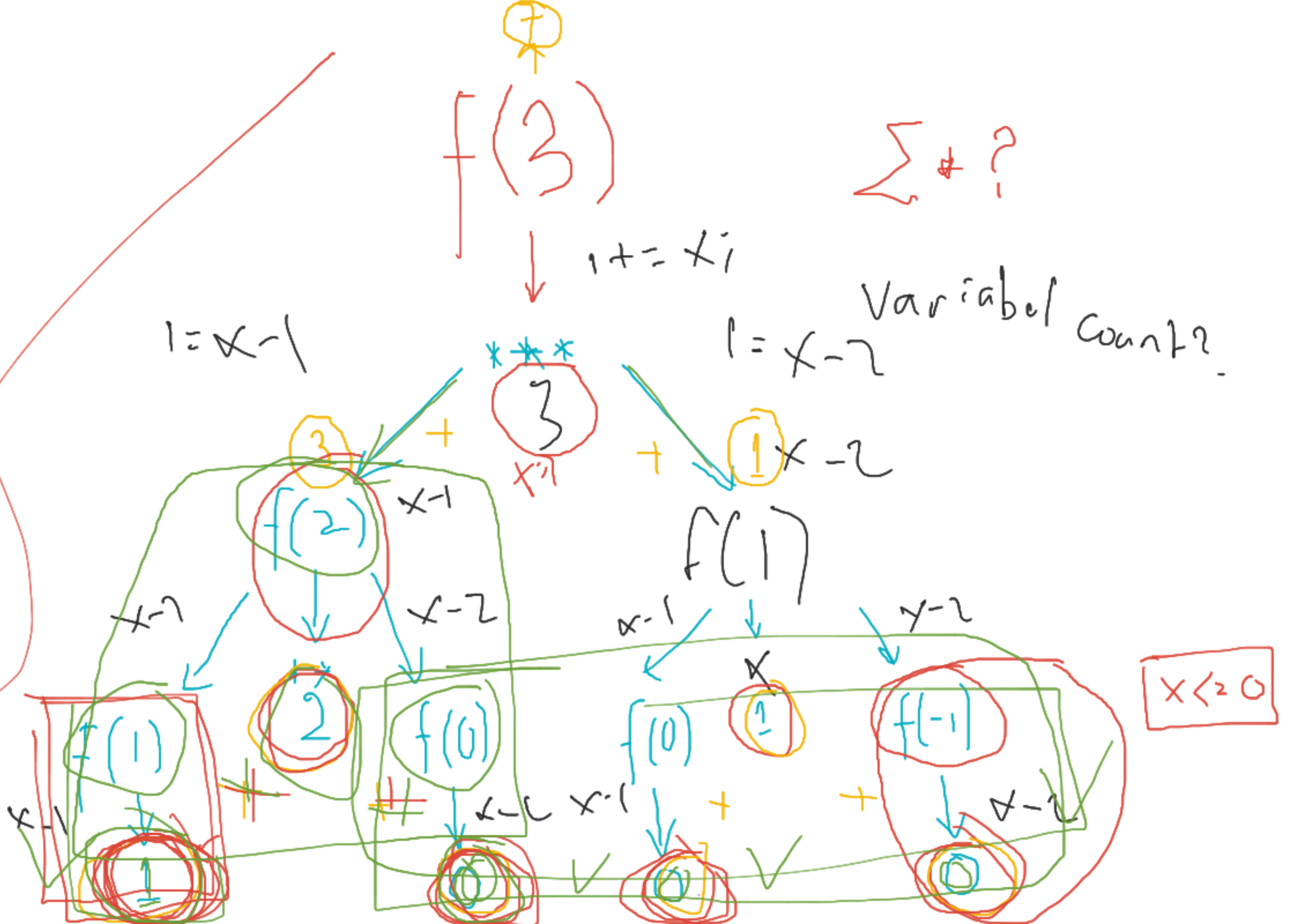
f(x-1) -

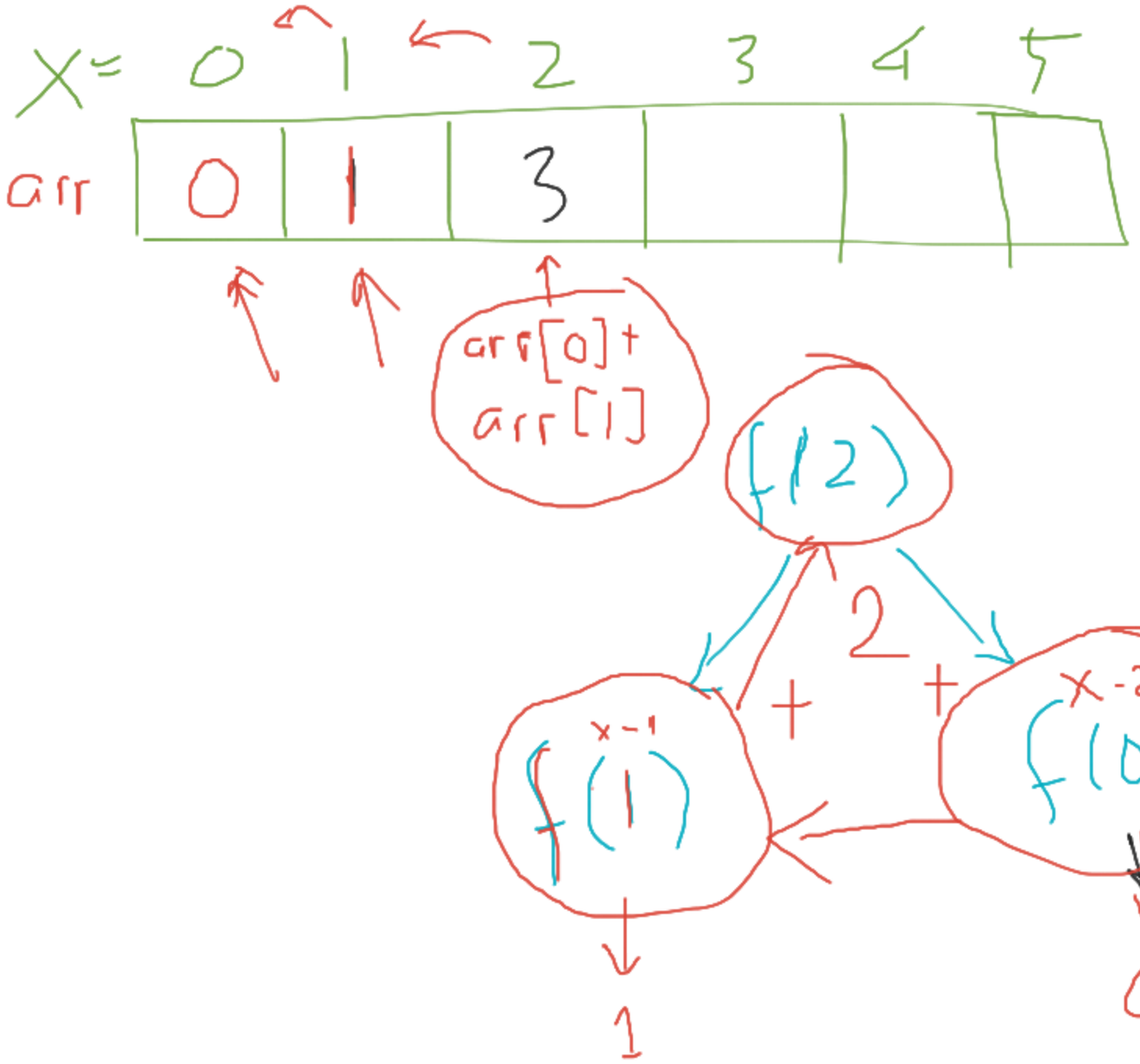
print(x) x 2

f(x-2) -

}

```
int fraktal_B(int x) {
  int i;
  if (x <= 0) return 0;
  else {
    int bintang = x;
    bintang += fraktal_B(x - 1);
    bintang += fraktal_B(x - 2);
    return bintang;
  }
}
```





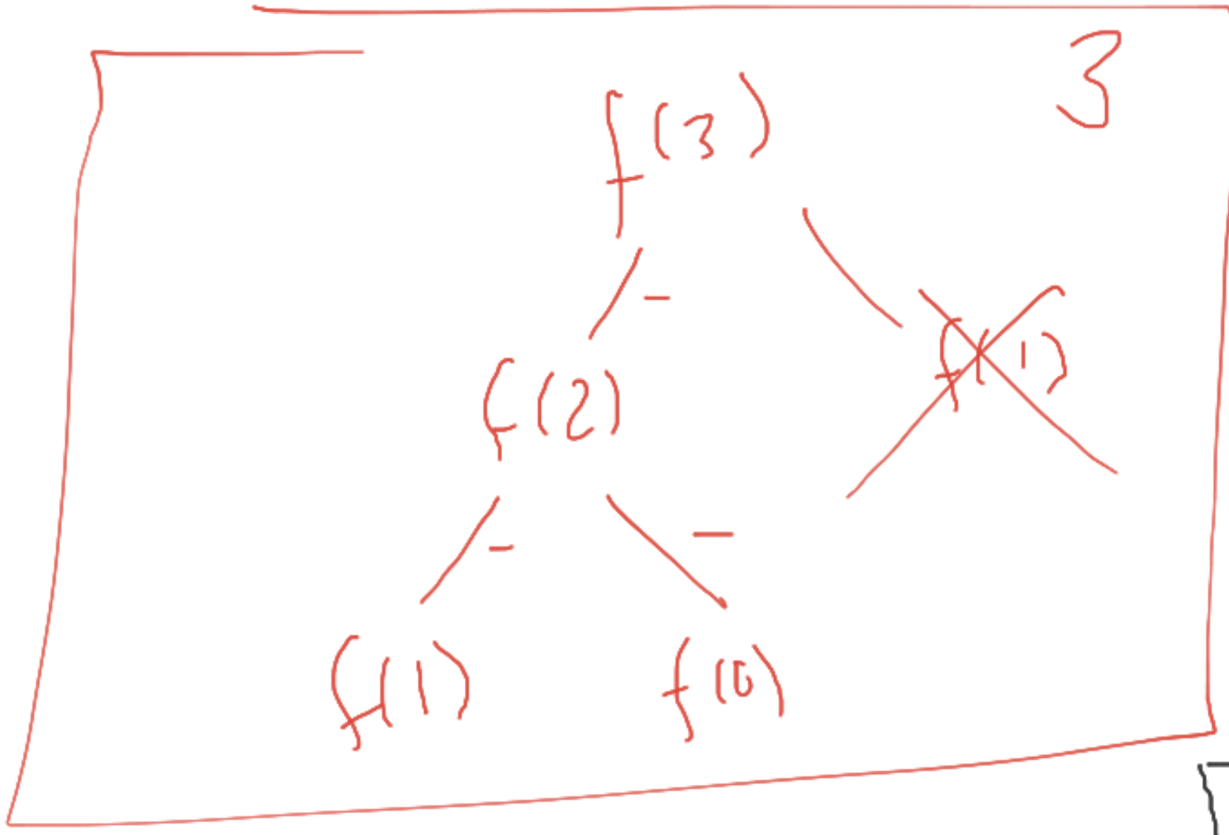
memo[x]:

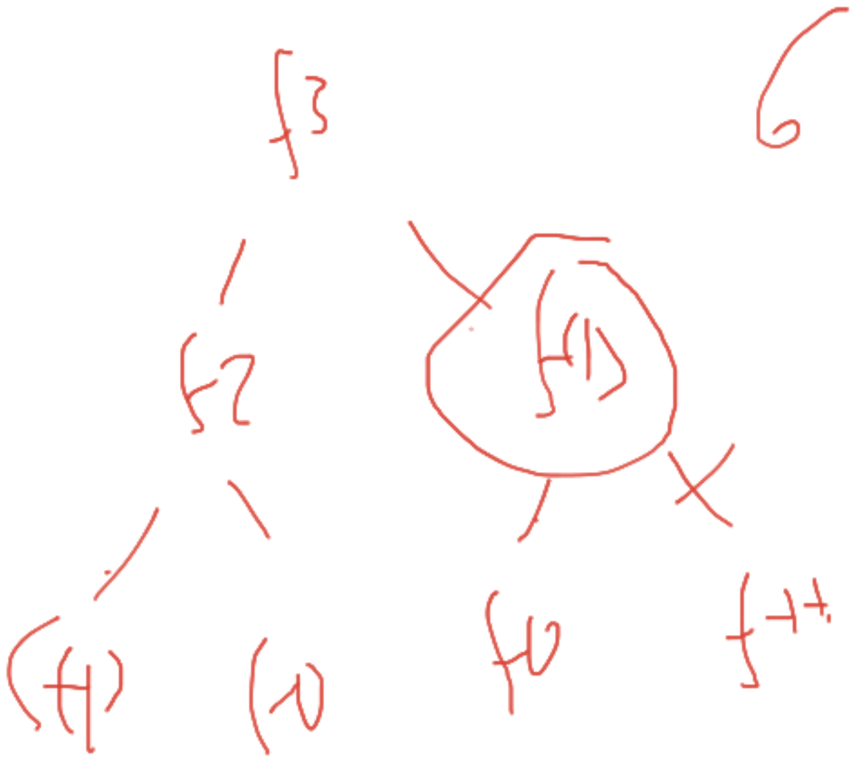
$f(2) = f(1) + f(0)$

$f(2) = arr[0] + arr[1]$

if(arr[x] != 0)
return arr[x]

else {
f(x)
}





Dynamic Programming !

Memorization

80