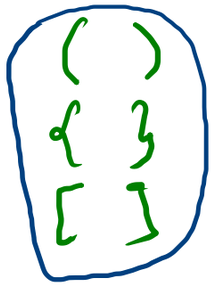
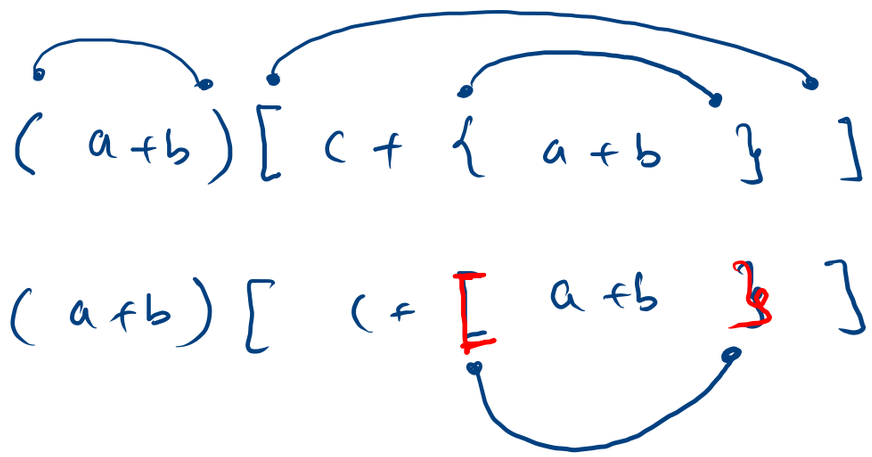
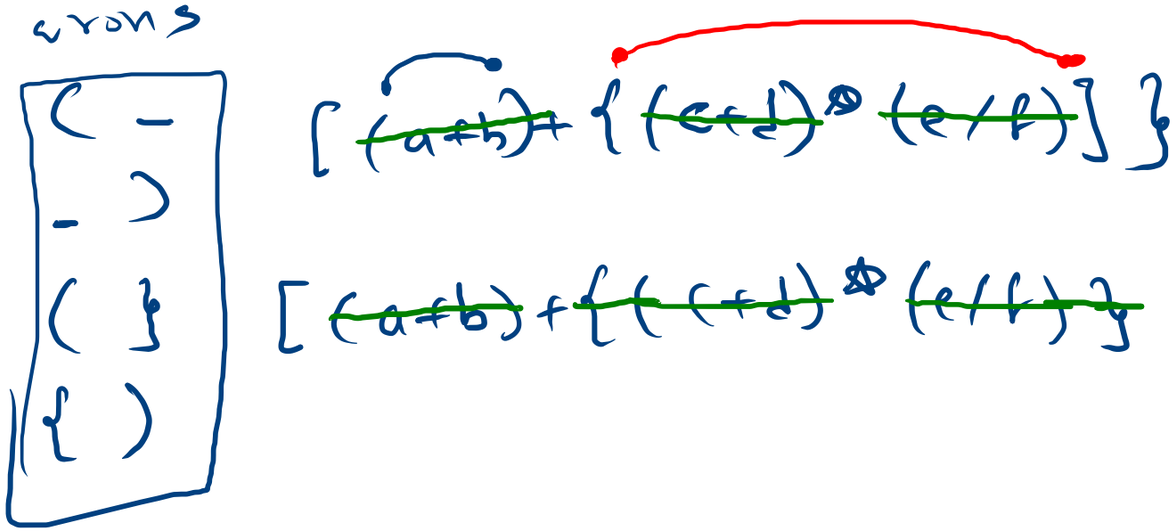


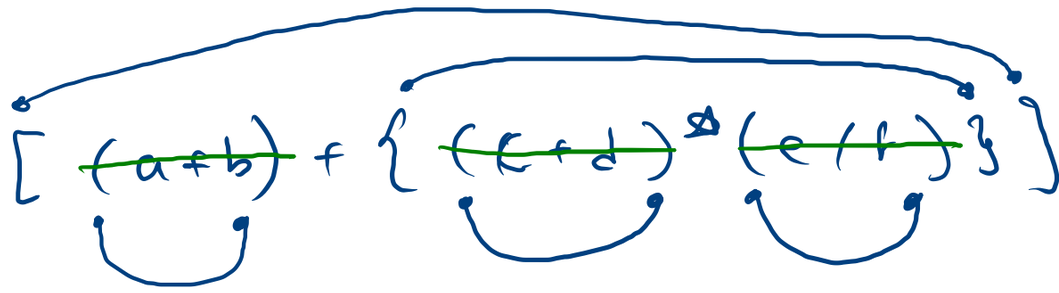
(a+b)
( )



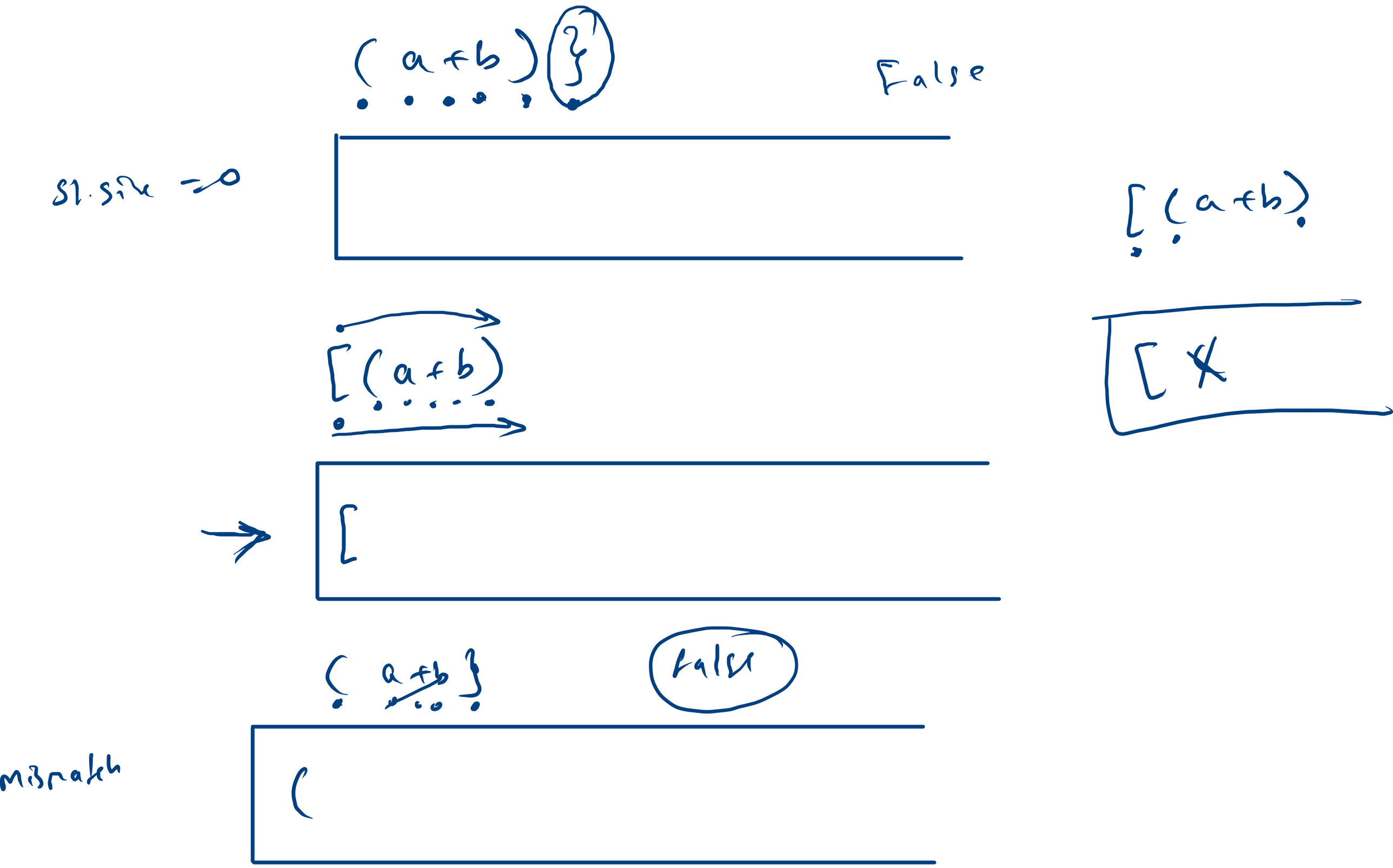
[(a + b) + {(c + d) * (e / f)}] -> true
[(a + b) + {(c + d) * (e / f)]} -> false
[(a + b) + {(c + d) * (e / f)} -> false
([(a + b) + {(c + d) * (e / f)]] -> false

errors



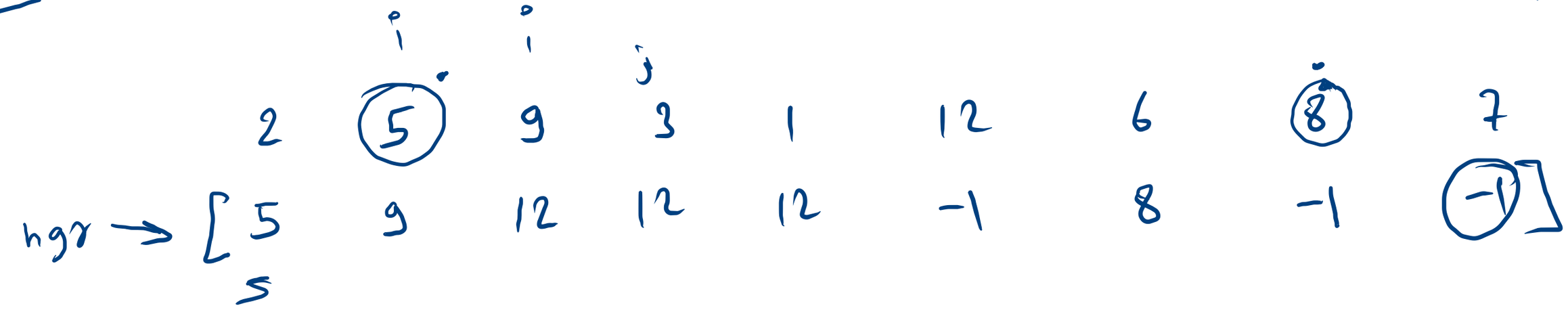


[(a + b) + {(c + d) * (e / f)}] -> true
[(a + b) + {(c + d) * (e / f)]} -> false
[(a + b) + {(c + d) * (e / f)} -> false
([(a + b) + {(c + d) * (e / f)}] -> false


(a+b)}
False
st.size = 0
[(a+b)
[(a+b)
[
[X
(a+b}
false
mismatch
(

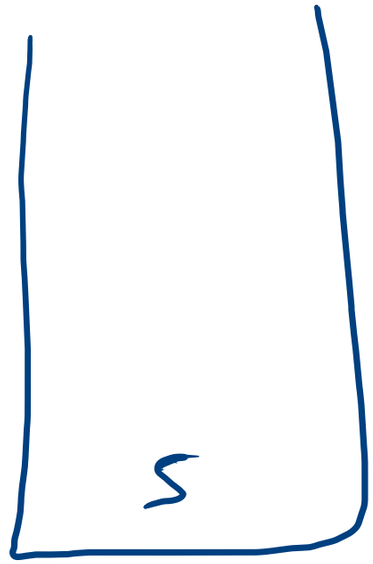
ngr

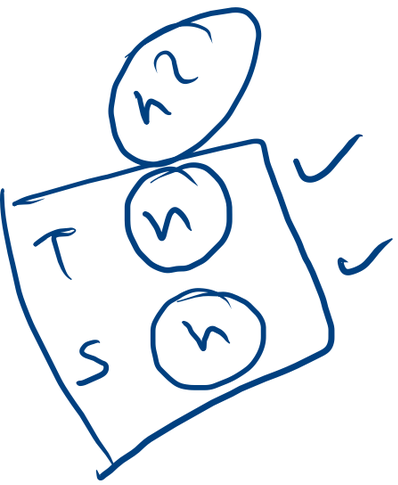
		i	i	j					
	2	5	9	3	1	12	6	8	7
ngr →	[5	9	12	12	12	-1	8	-1	-1]
	s								

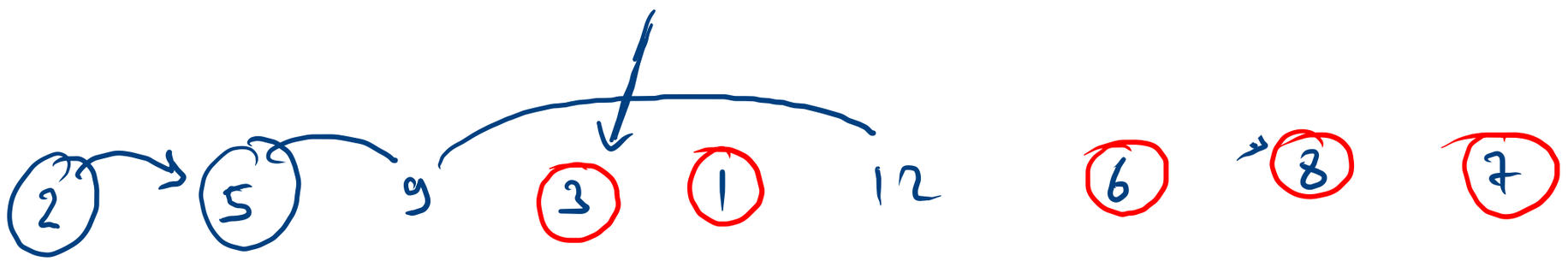


$j = i+1$

s



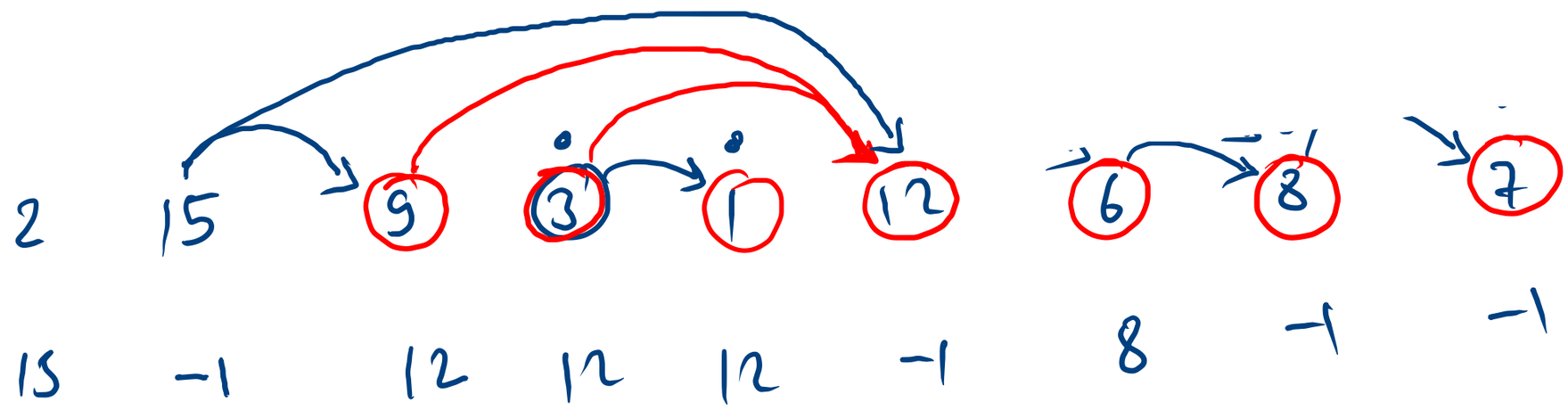




5 9 12 12 12 -1 8 -1 -1

blue ✓

red ✓

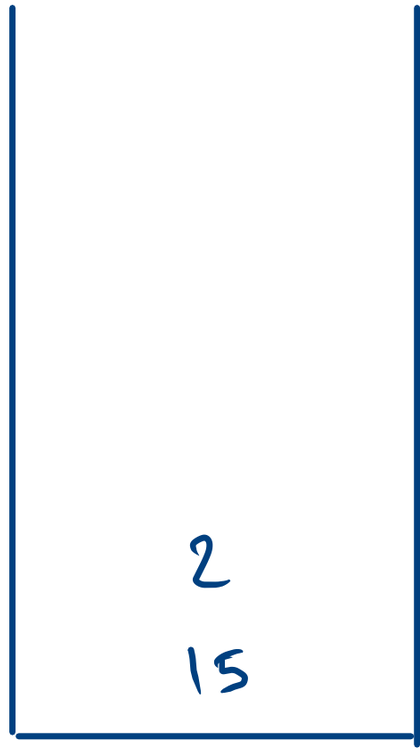


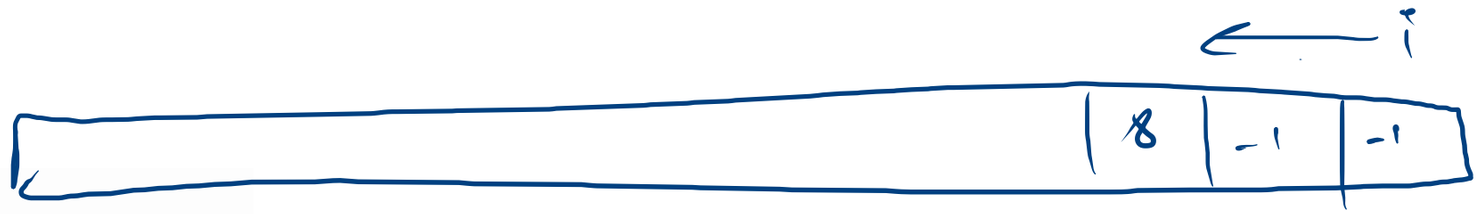
15 <= 2
while (st.peek <= val)
st.pop

ans

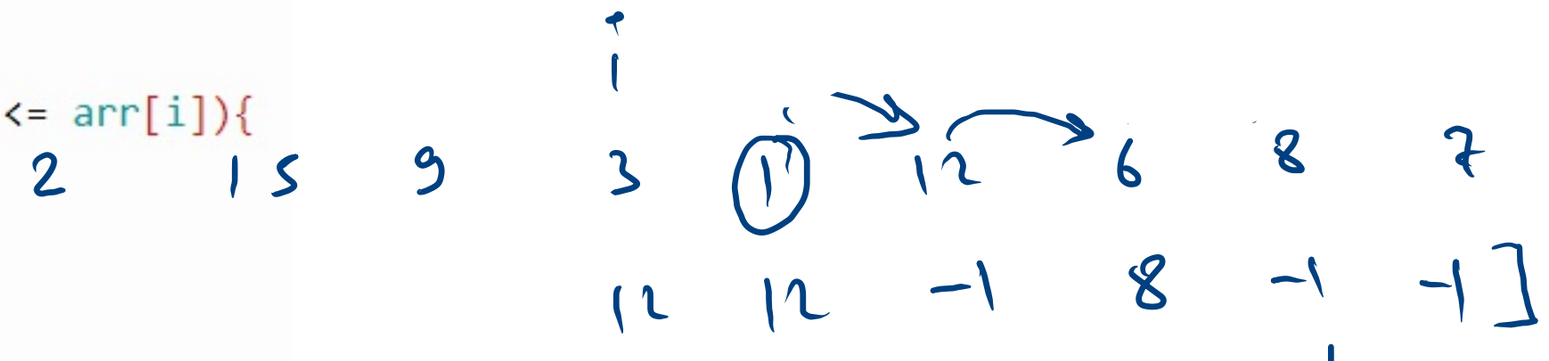
+ push

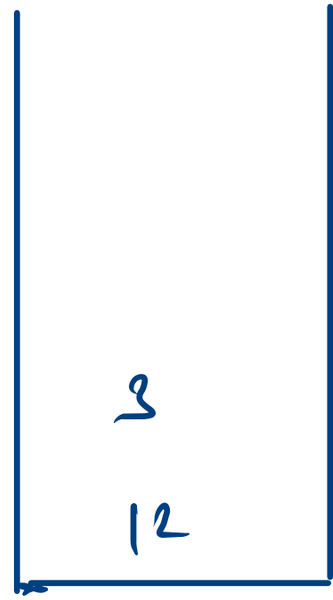
blue →

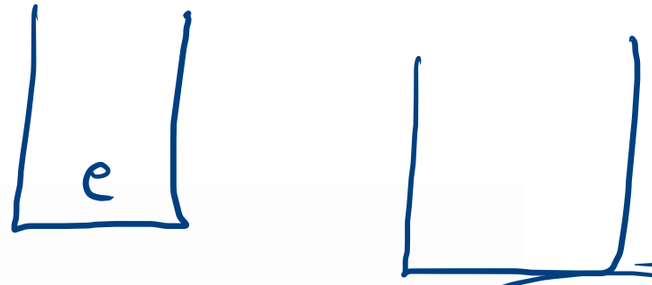





```java
int n = arr.length;

int ngr[] = new int[n];
Stack<Integer> st = new Stack<>();
for(int i=n-1; i>=0;i--){
    while(st.size() >0 && st.peek() <= arr[i]){
        st.pop();
    }
    int ans=-1;
    if(st.size()>0){
        ans = st.peek();
    }
    ngr[i] = ans;

    st.push(arr[i]);
}

return ngr;
```







n elem

```
int n = arr.length;

int ngr[] = new int[n];
Stack<Integer> st = new Stack<>();
for(int i=n-1; i>=0;i--){
    while(st.size() >0 && st.peek() <= arr[i]){
        st.pop();
    }
    int ans=-1;
    if(st.size()>0){
        ans = st.peek();
    }
    ngr[i] = ans;

    st.push(arr[i]);
}

return ngr;
```

n times

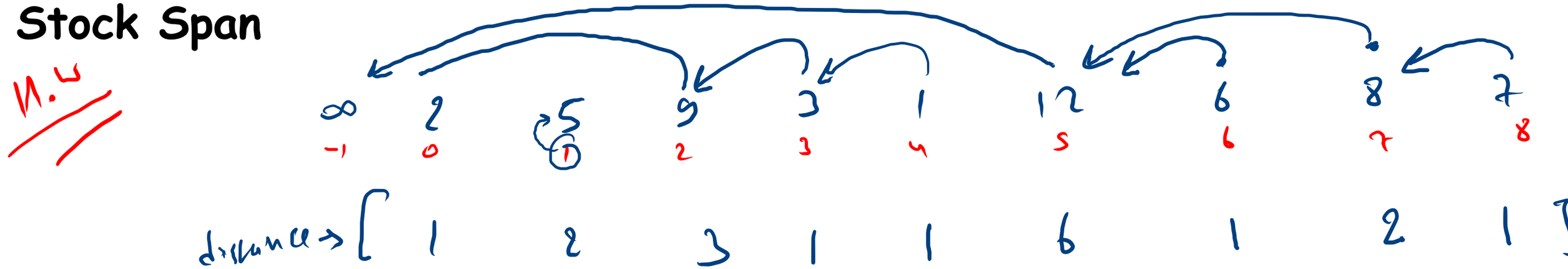
$\boxed{0+1+a \ldots z \leq n}$

n

$O(n)$ $O(n) + O(n)$

i	work
n-1	0
n-2	1
n-3	0
n-4	2
n-5	a
n-6	b
⋮	⋮
0	z

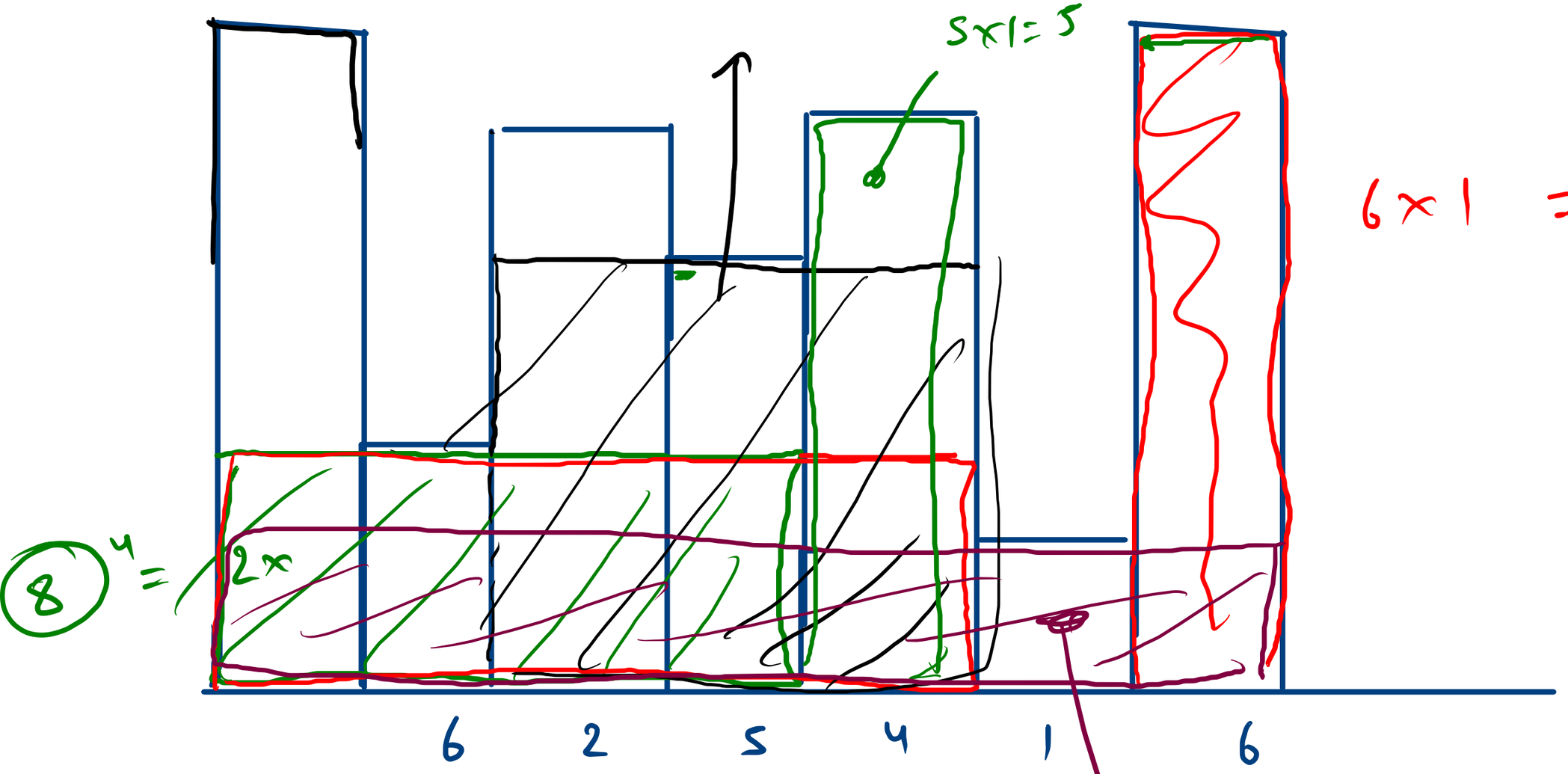
Stock Span



for the array [2 5 9 3 1 12 6 8 7]
span for 2 is 1
span for 5 is 2
span for 9 is 3
span for 3 is 1
span for 1 is 1
span for 12 is 6
span for 6 is 1
span for 8 is 2
span for 7 is 1

4 x 3 = (12)

5 x 1 = 5



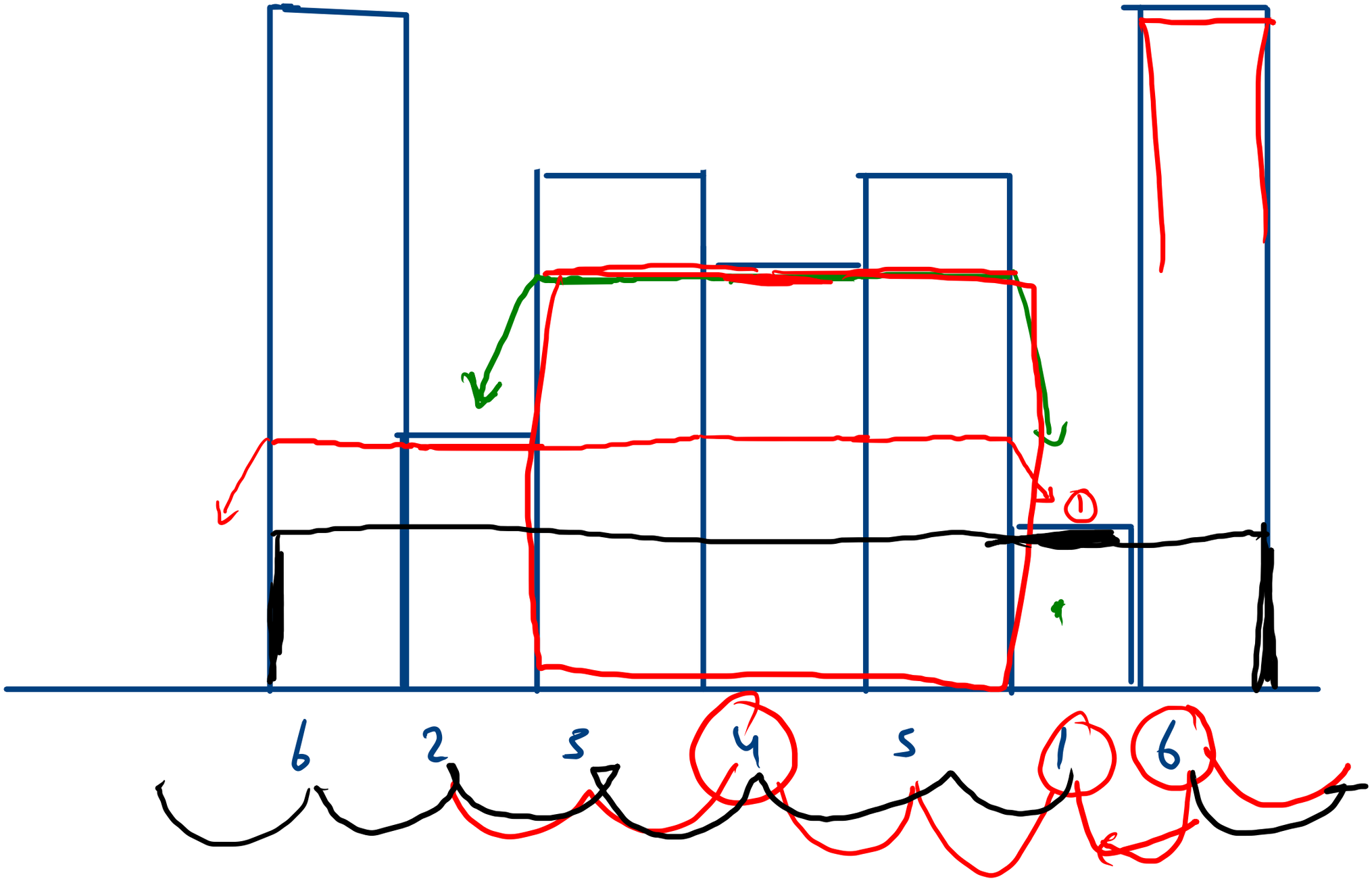
6 x 1 = 6

2 x 5 = 10

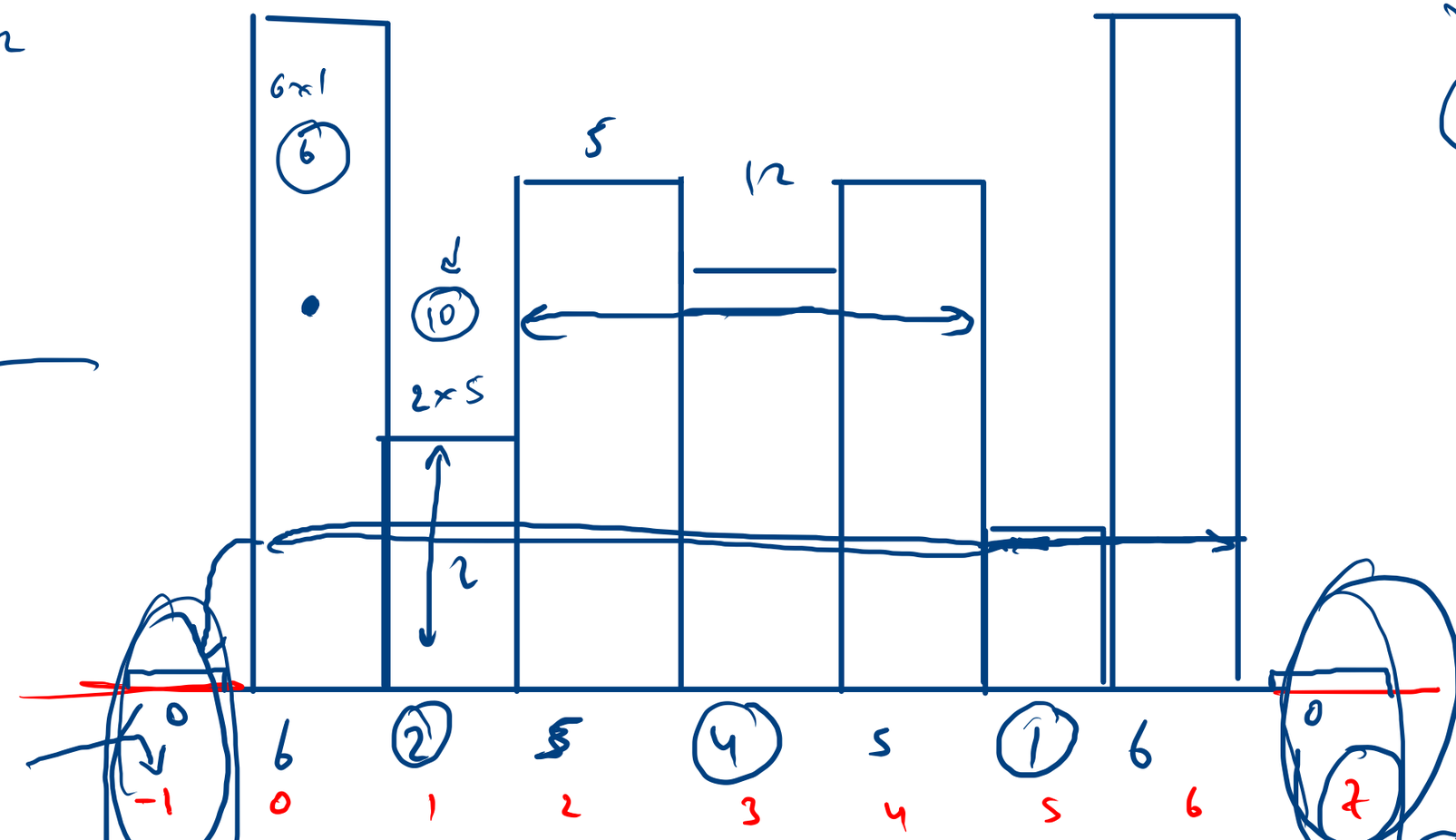
1 x 7 = (7)

dl

(12)


1
6
2
5
4
5
1
6

max = ~~6~~ ~~10~~ 12



Left small	-1	-1	1	1	3	-1	5
right small	1	5	3	5	5	7	7

$\gamma 5 - L5 - 1$

$5 - (-1) - 1$

$5 + 1 - 1$

$5 - 1 - 1$

3

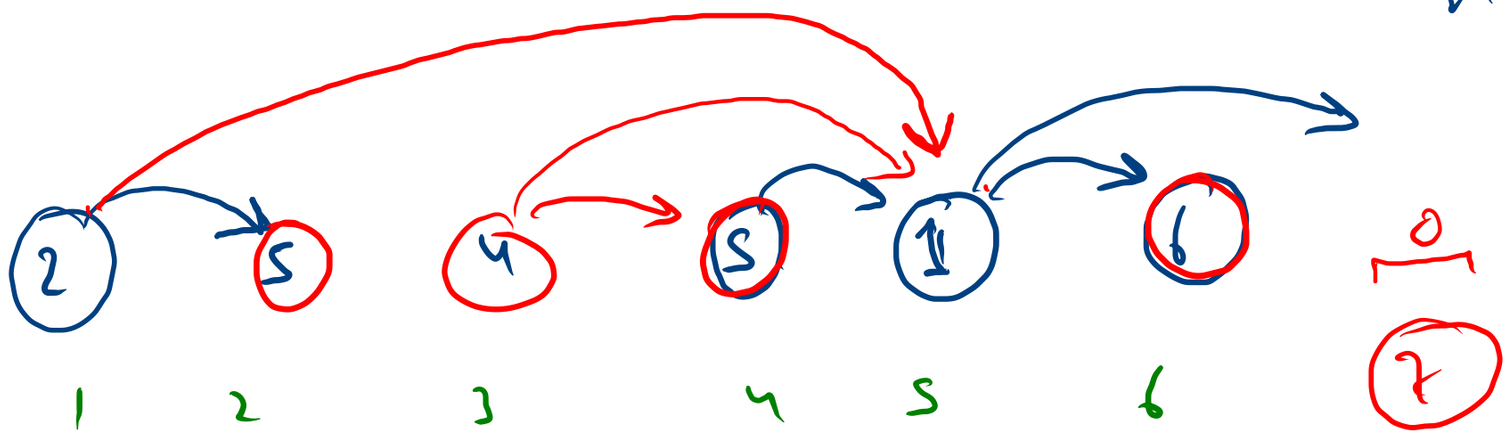
$7 - (-1) - 1$

$7 + 1 - 1$

7

small

class = 7



5 3 5 5 7 7]

nsr [

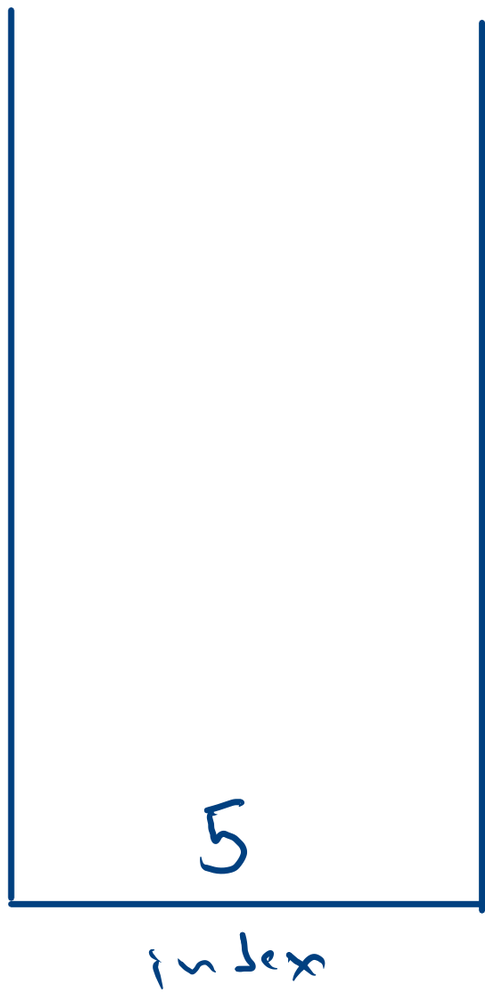
$s \geq 2$

$a[peek] \geq val$

pop

anr

push(i)



k=4

8

9

9

2 9 3 8 1 7 12 6 14 4 32 0 7 19 8 12 6

9 9 8 12 12 14 14 32 32 32 32 19 19 19

(n-k) * k

5 5 5 5 5 5 5 5 5 5

k = 4

ws = 0
we = k-1

we < h

hgr	1	6	3	6	5	6	8	8	10	10	17	12	13	17	15	17	17	
	2	9	3	8	1	7	12	6	14	4	32	0	7	19	8	12	6	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
															ws			we

g

k = 5
6

17 ≤ 14

if (g < ws)
g = ws

g	9	8	12	12	14	14	32	32	19	19	19		

while (hgr[g] ≤ we)
g = hgr[g]
min (ans[5])

k = 3

ws 0 1 2 3
we 2 3 4 5
g 0 1 2 3 4

ngr

i	0	1	2	3	4	5	6	7
	2	9	3	8	1	7	12	

ws we g

```java
int ngr[] = nextGreaterRightIndex(a);
int g = 0;
for(int ws=0, we=k-1; we<n; ws++, we++){
    if(g<ws){
        g=ws;
    }
    while(ngr[g] <= we){
        g = ngr[g];
    }

    System.out.println(a[g]);
}
```

4 <= 4

g g 71