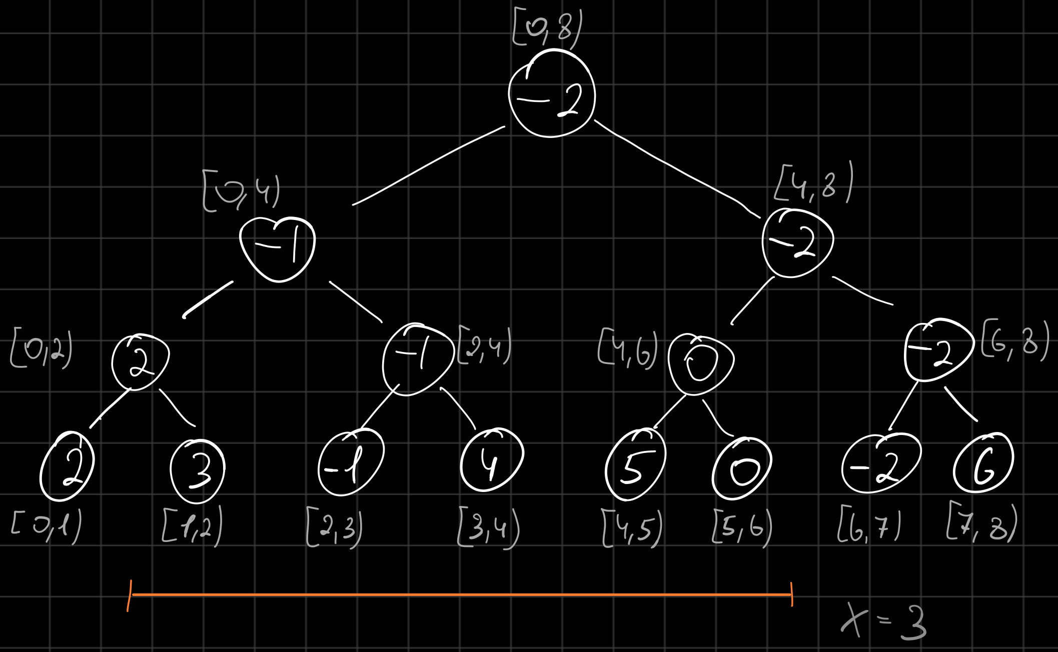
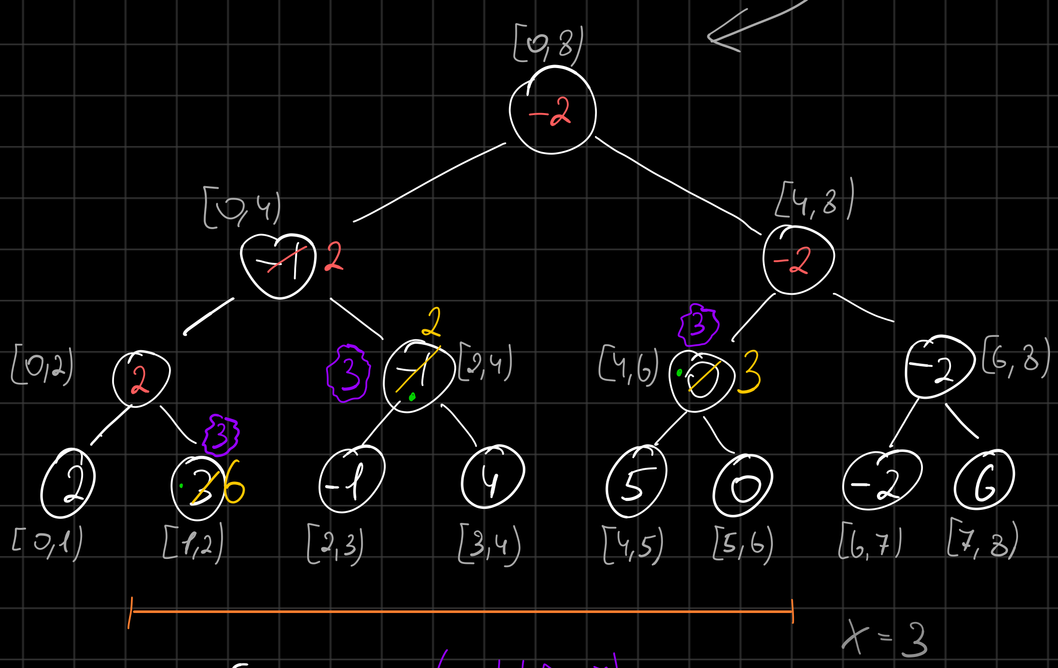
# Дерево отрезков

· get $\ell$ $r$ // min$(a_\ell, a_{\ell+1}, \dots, a_{r-1})$

· add $\ell$ $r$ $x$ // $a_\ell += x$, $a_{\ell+1} += x, \dots,$ $a_{r-1} += x$



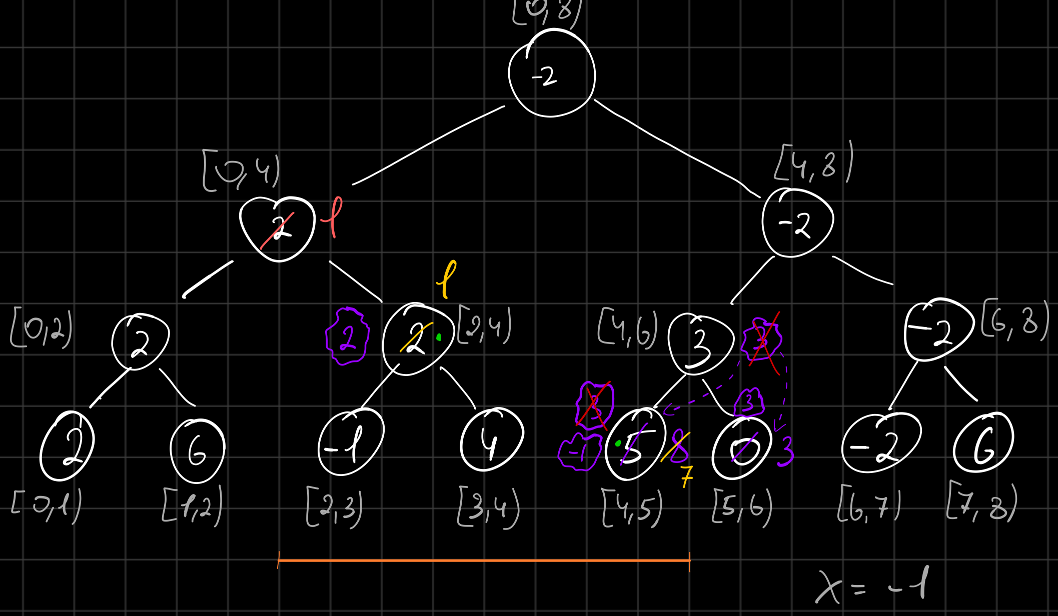
исп. ленивые/отложенные операции



▨ – хотим к детям прибавить x (add[v])

▨ – меняем предков

▨ – прибавл. к вершинам x



Инварианты:

1) На выхде знач-е корректное

2) add[v] хранит, сколько прибавить ко всем детям (не считая текущ. вершину)

```
       где      куда
add(v, l, r, ql, qr, x)
    if (qr ≤ l || r ≤ ql):
        return
    if (ql ≤ l && r ≤ qr):
        tree[v] += x
        add[v] += x
        return
    m = (l+r)/2
    push(v)
    add(2v+1, l, m, ql, qr, x)
    add(2v+2, m, r, ql, qr, x)
    tree[v] = min(tree[2v+1], tree[2v+2])
```

```
push(v)
    if (add[v] == 0): return
    tree[2v+1] += add[v]
    tree[2v+2] += add[v]
    add[2v+1] += add[v]
    add[2v+2] += add[v]
    add[v] = 0
```

```
get(v, l, r, ql, qr):
    if (qr ≤ l || r ≤ ql):
        return +∞
    if (ql ≤ l && r ≤ qr):
        return tree[v]
    push(v)
    m = (l+r)/2
    return min(get(2v+1, l, m, ql, qr), get(2v+2, m, l, ql, qr))
```

- get $l$ $r$ // sum$(a_l, a_{l+1}, \dots, a_{r-1})$
- add $l$ $r$ $x$

```
get(v, l, r, ql, qr):
    if (qr ≤ l || r ≤ ql):
        return 0
    if (ql ≤ l && r ≤ qr):
        return tree[v]
    push(v)
    m = (l+r)/2
    return sum(get(2v+1, l, m, ql, qr), get(2v+2, m, l, ql, qr))
```

```
          где        куда
add(v, l, r, ql, qr, x)
    if (qr ≤ l || r ≤ ql):
        return
    if (ql ≤ l && r ≤ qr):
        tree[v] += x · (r - l)
        add[v] += x
        return
    m = (l + r) / 2
    push(v, l, m, r)
    add(2v+1, l, m, ql, qr, x)
    add(2v+2, m, r, ql, qr, x)
    tree[v] = min(tree[2v+1], tree[2v+2])
```

```
push(v, l, m, r)
    if (add[v] == 0): return
    tree[2v+1] += add[v] · (m - l)
    tree[2v+2] += add[v] · (r - m)
    add[2v+1] += add[v]
    add[2v+2] += add[v]
    add[v] = 0
```

- get $l$ $r$ //sum$(a_l, a_{l+1}, \dots, a_{r-1})$
- add $l$ $r$ $x$ // $a_l = x, a_{l+1} = x, \dots, a_{r-1} = x$

```
get(v, l, r, ql, qr):
    if (qr ≤ l || r ≤ ql):
        return 0
    if (ql ≤ l && r ≤ qr):
        return tree[v]
    push(v)
    m = (l + r) / 2
    return sum(get(2v+1, l, m, ql, qr), get(2v+2, m, l, ql, qr))
```

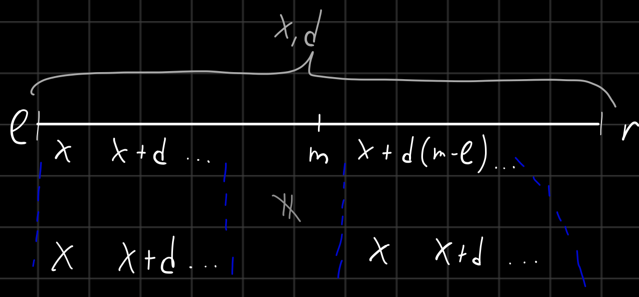
```
          где     куда
add(v, l, r, ql, qr, x)
    if (qr <= l || r <= ql):
        return
    if (ql <= l && r <= qr):
        tree[v] = x·(r-l)
        add[v] += x
        return
    m = (l+r)/2
    push(v, l, m, r)
    add(2v+1, l, m, ql, qr, x)
    add(2v+2, m, r, ql, qr, x)
    tree[v] = min(tree[2v+1], tree[2v+2])
```

```
push(v, l, m, r)
    if (add[v] == 0): return
    tree[2v+1] = add[v]·(m-l)
    tree[2v+2] = add[v]·(r-m)
    add[2v+1] += add[v]
    add[2v+2] += add[v]
    add[v] = 0
```

set x
set y  <=> set y

add l r x d // $a_l$ += x , $a_{l+1}$ += x+d , $a_{l+2}$ += x+2d, ..., $a_{r-1}$ += x+d·(...)

$$\left.\begin{array}{l} x_1 \quad d_1 \to x_1 \quad x_1+d_1 \quad x_1+2d_1 \quad \dots \\ x_2 \quad d_2 \to x_2 \quad x_2+d_2 \quad x_2+2d_2 \quad \dots \end{array}\right\} \Rightarrow (x_1+x_2)\ (d_1+d_2)$$



А что не получится реализовать?

- get $l$ $r$ // $\min(a_l, a_{l+1}, \ldots, a_{r-1})$
- add $l$ $r$ $x$ //...