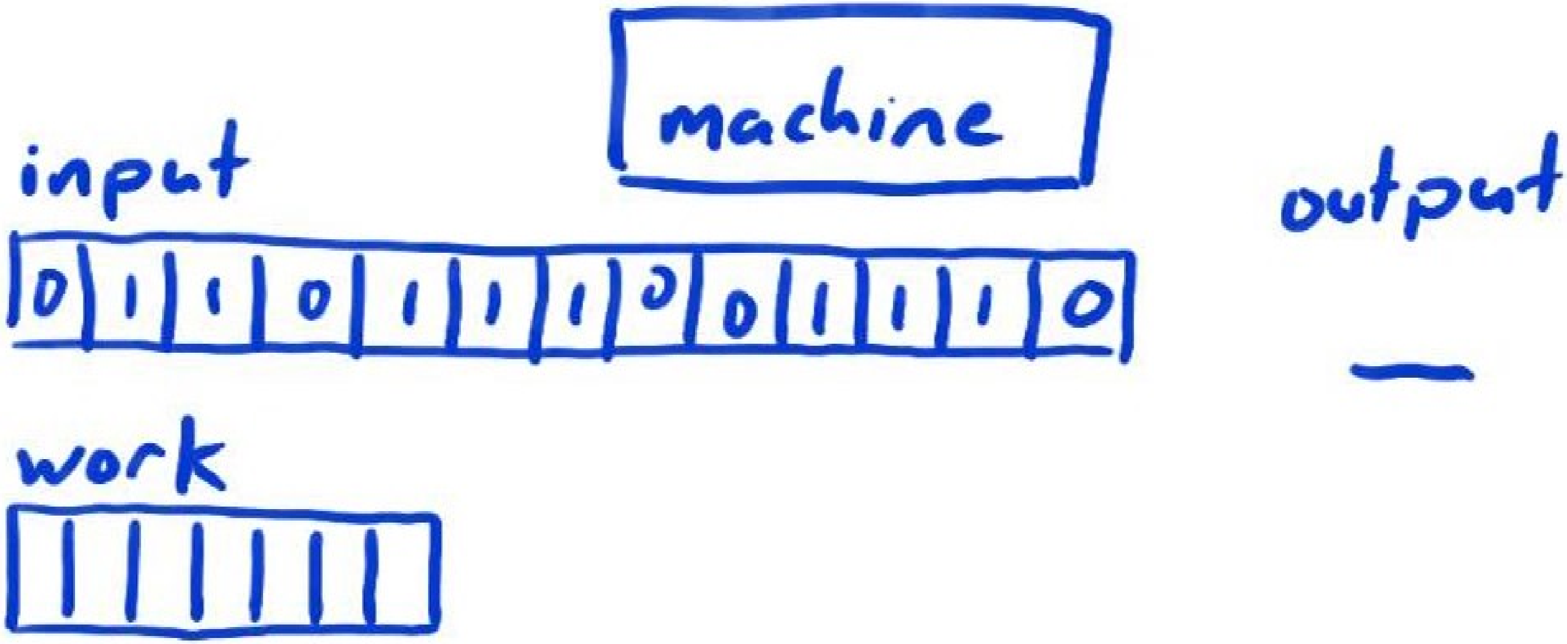
Catalytic Computing Between L and P

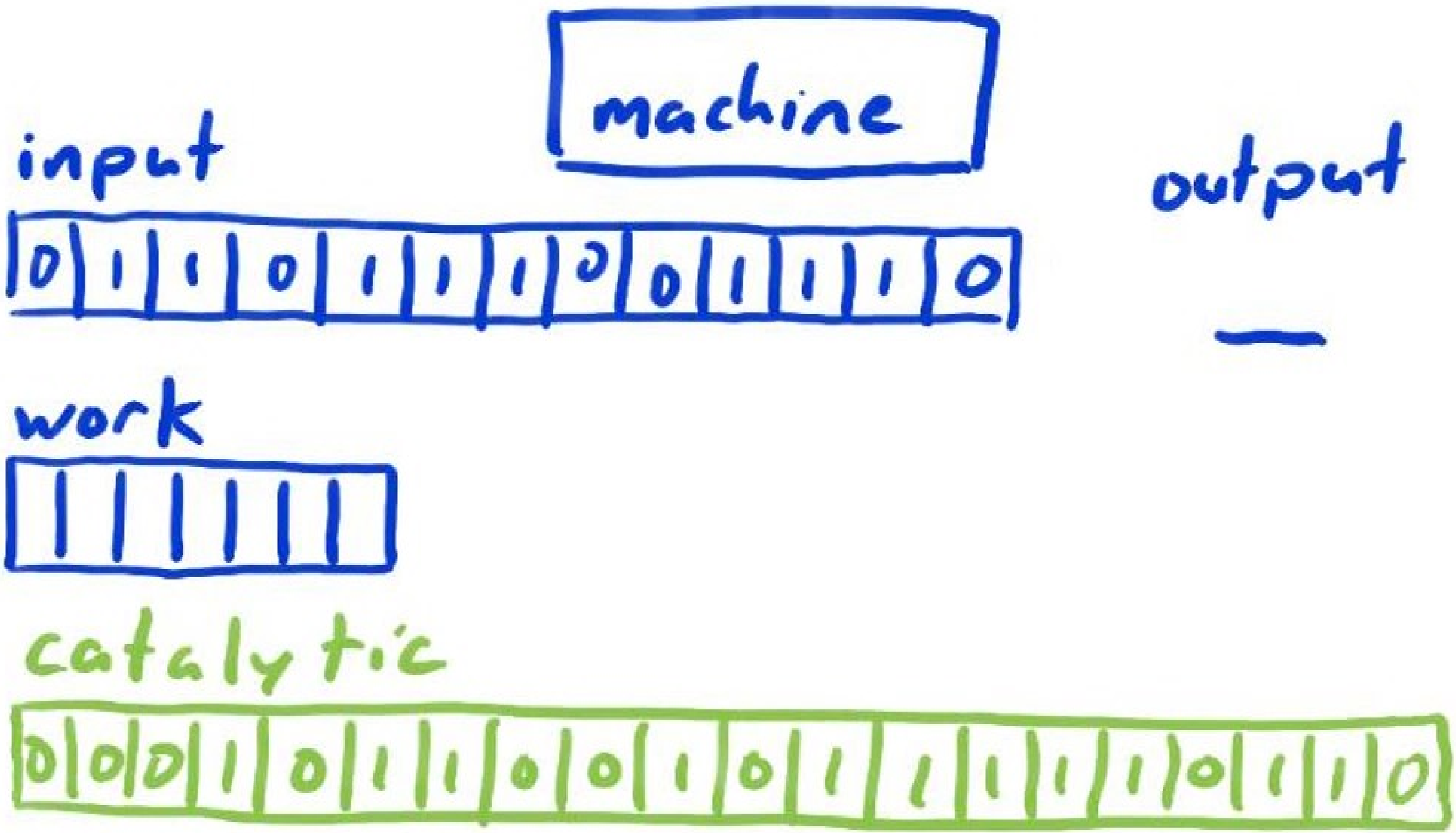
Ian Mertz

December 6, 2019

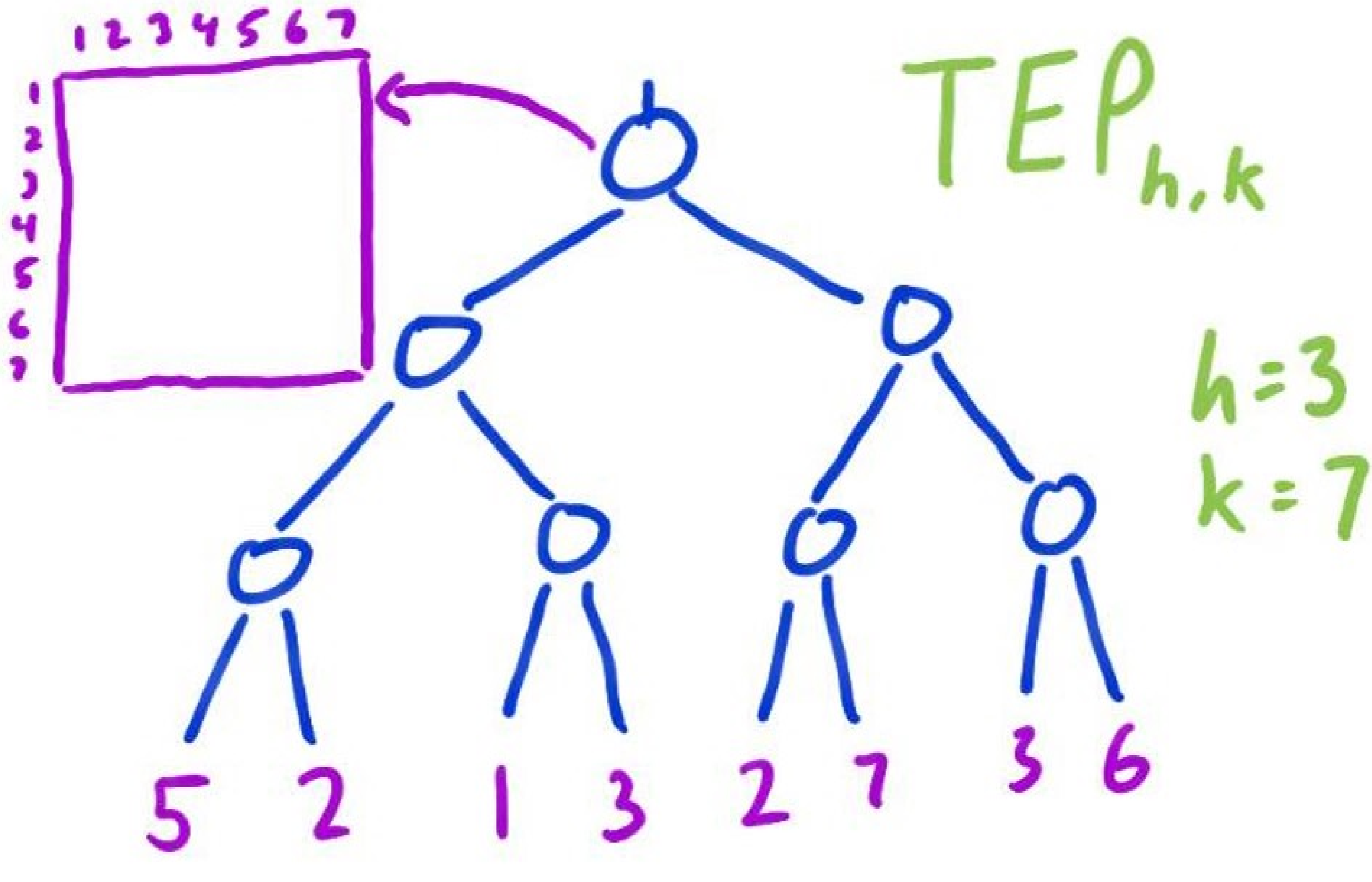
What is catalytic computing?



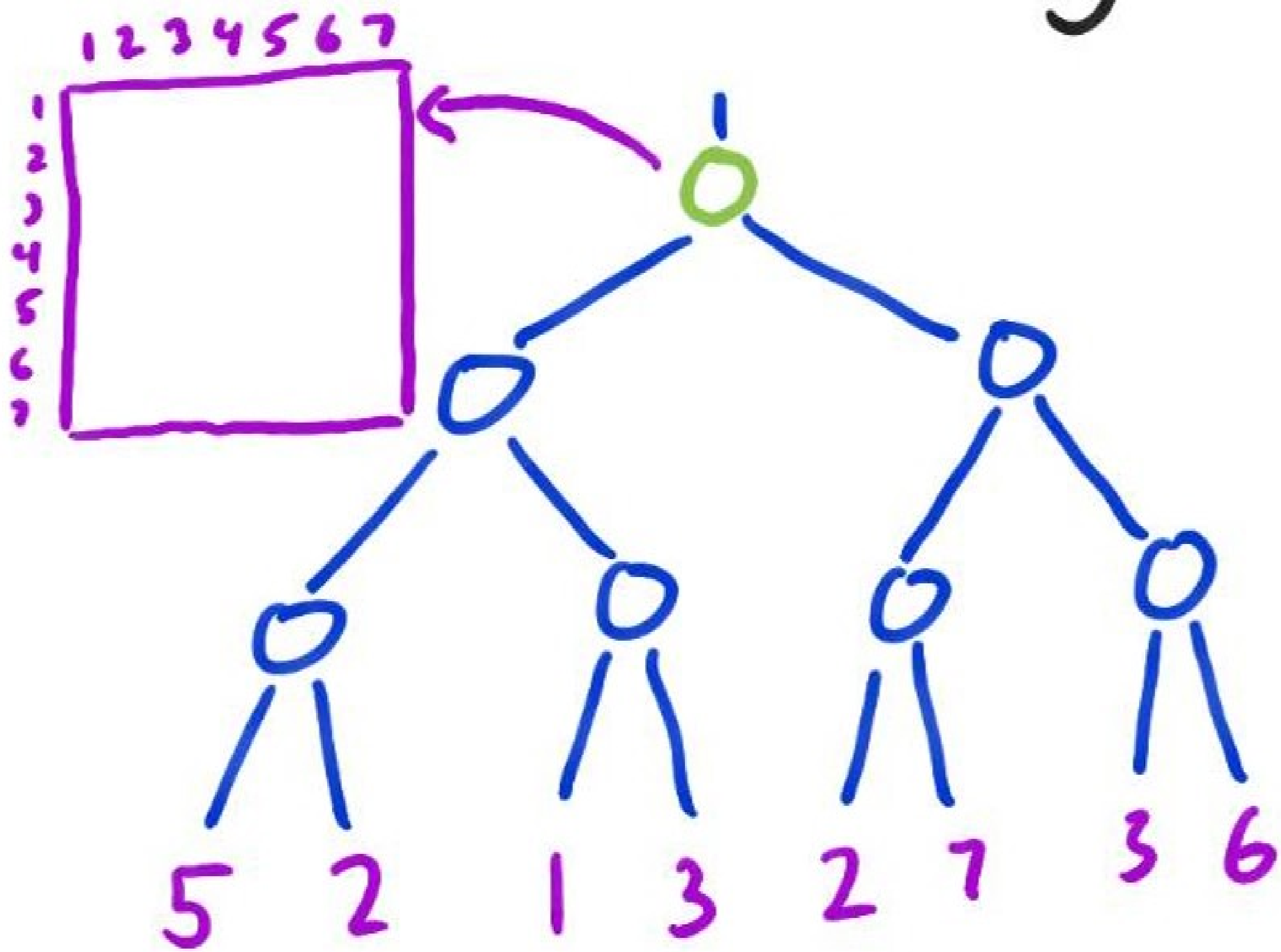
What is catalytic computing?



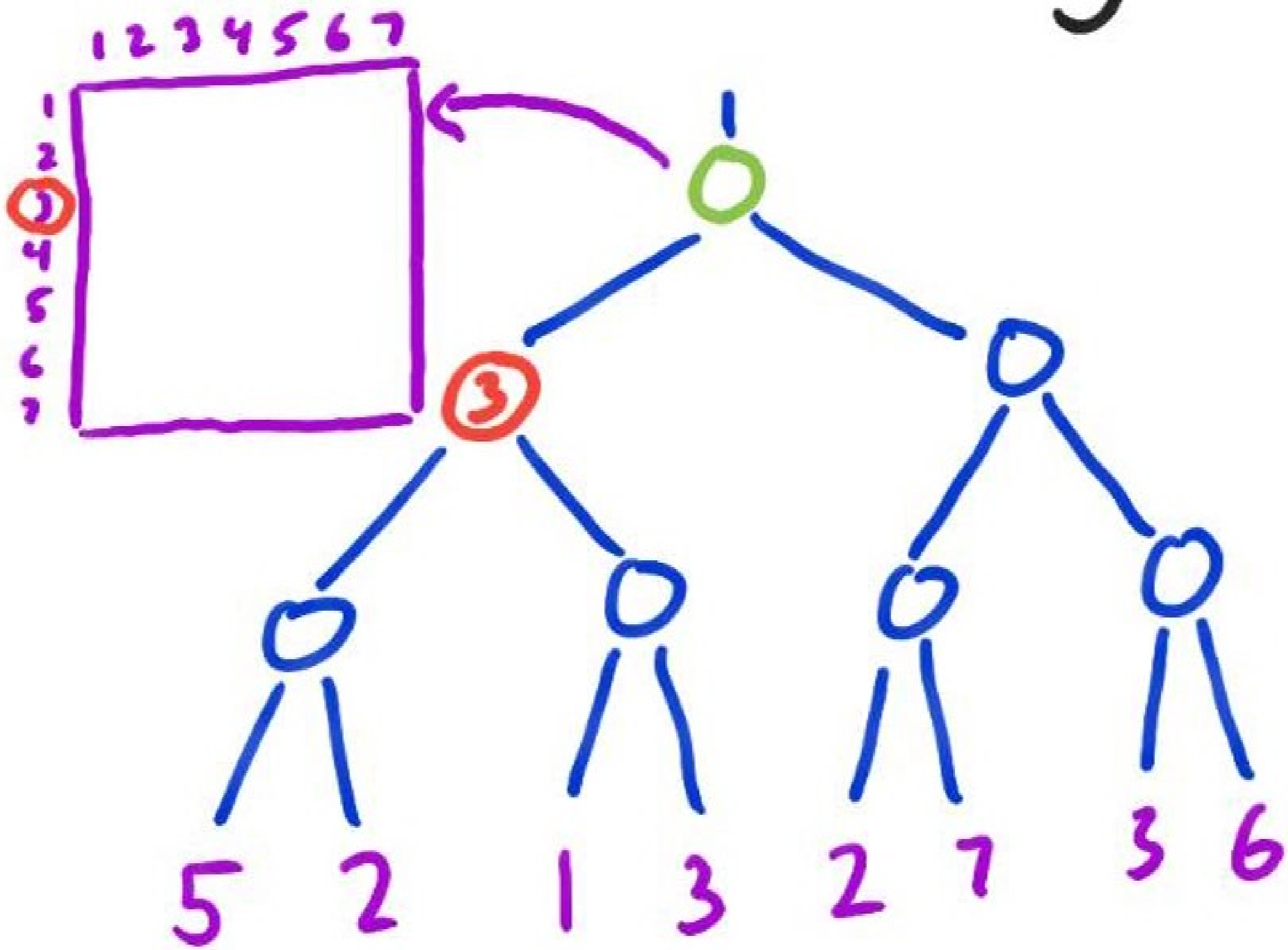
Tree Evaluation Problem



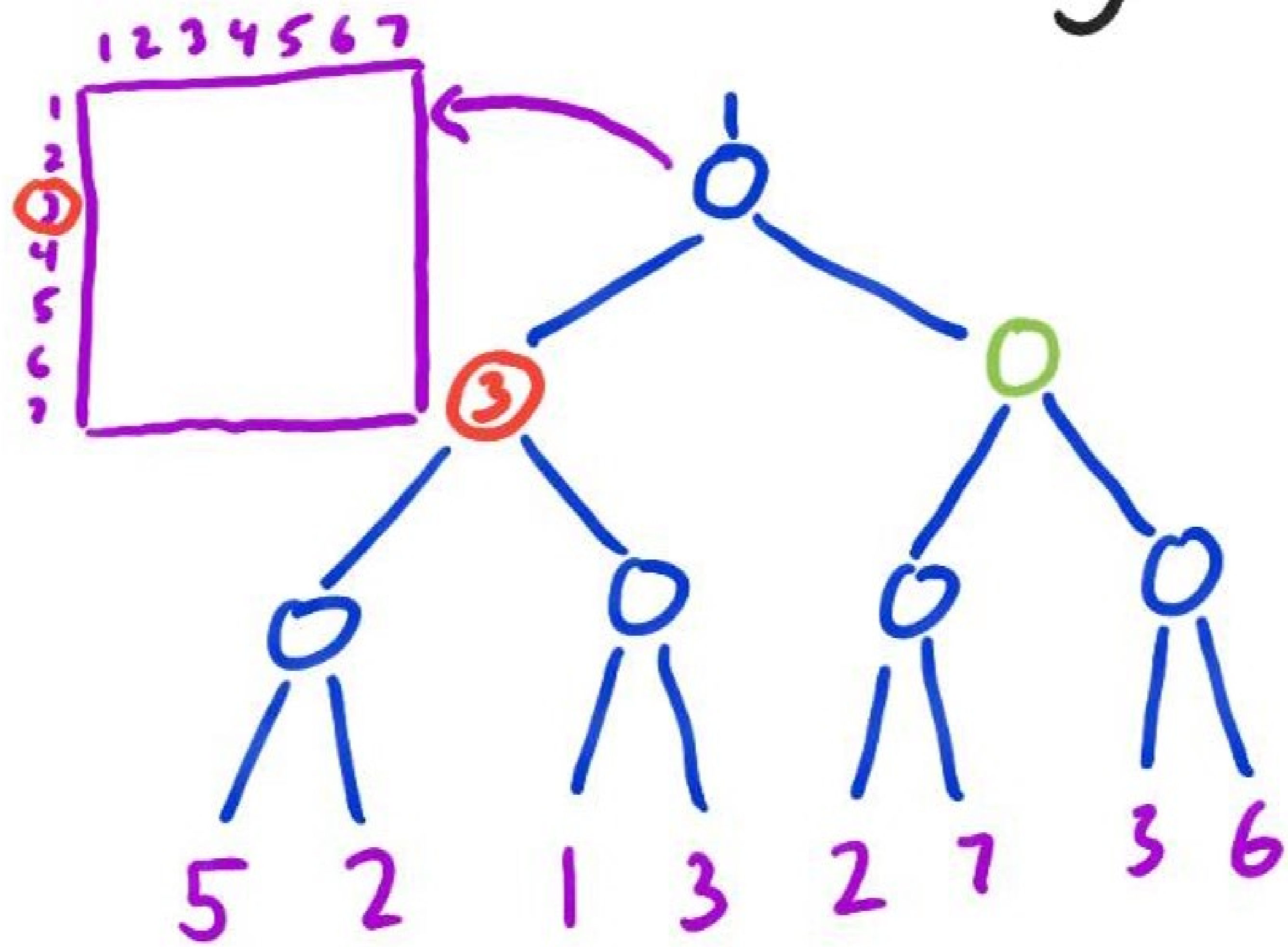
Pebbling



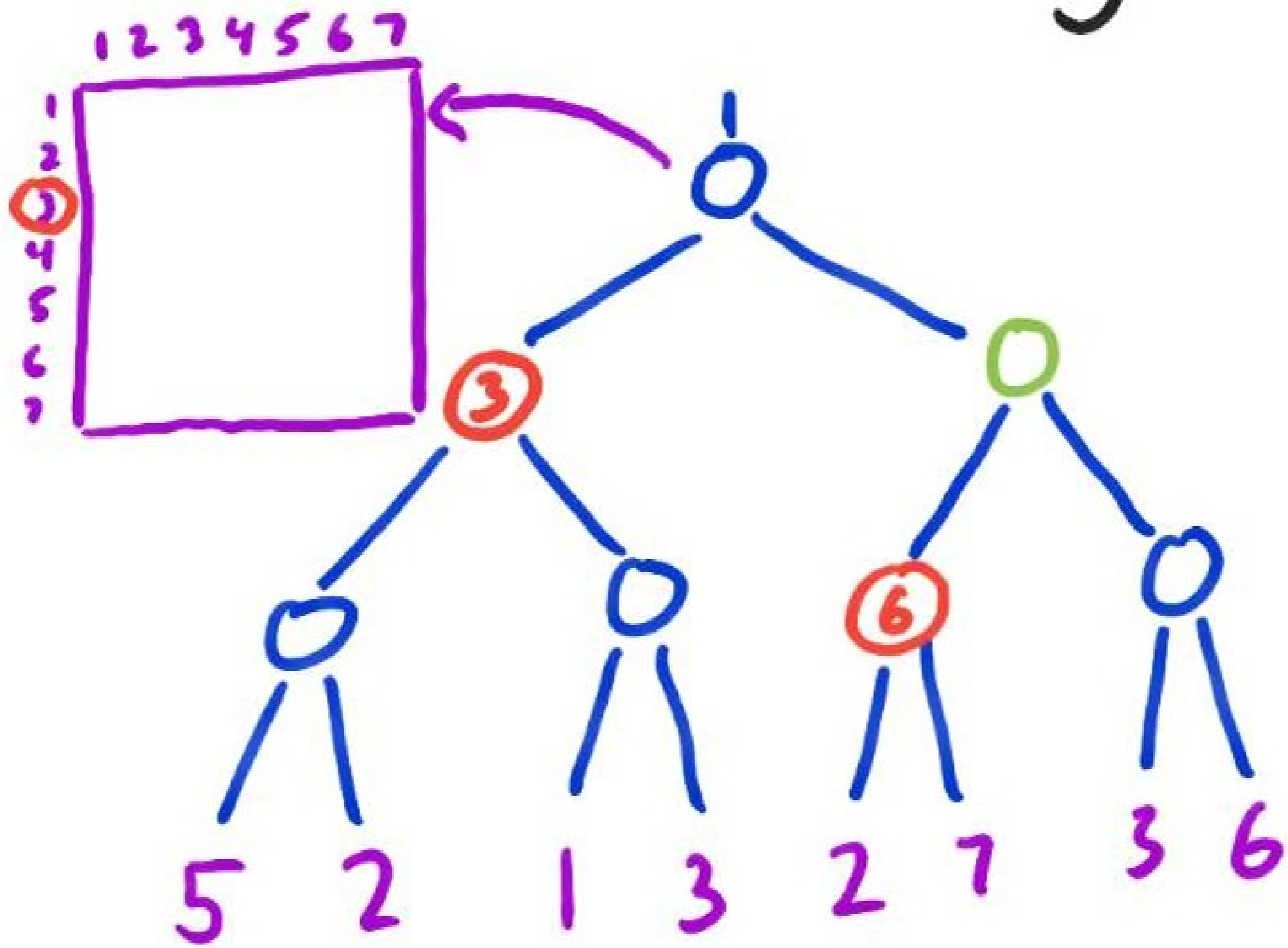
Pebbling



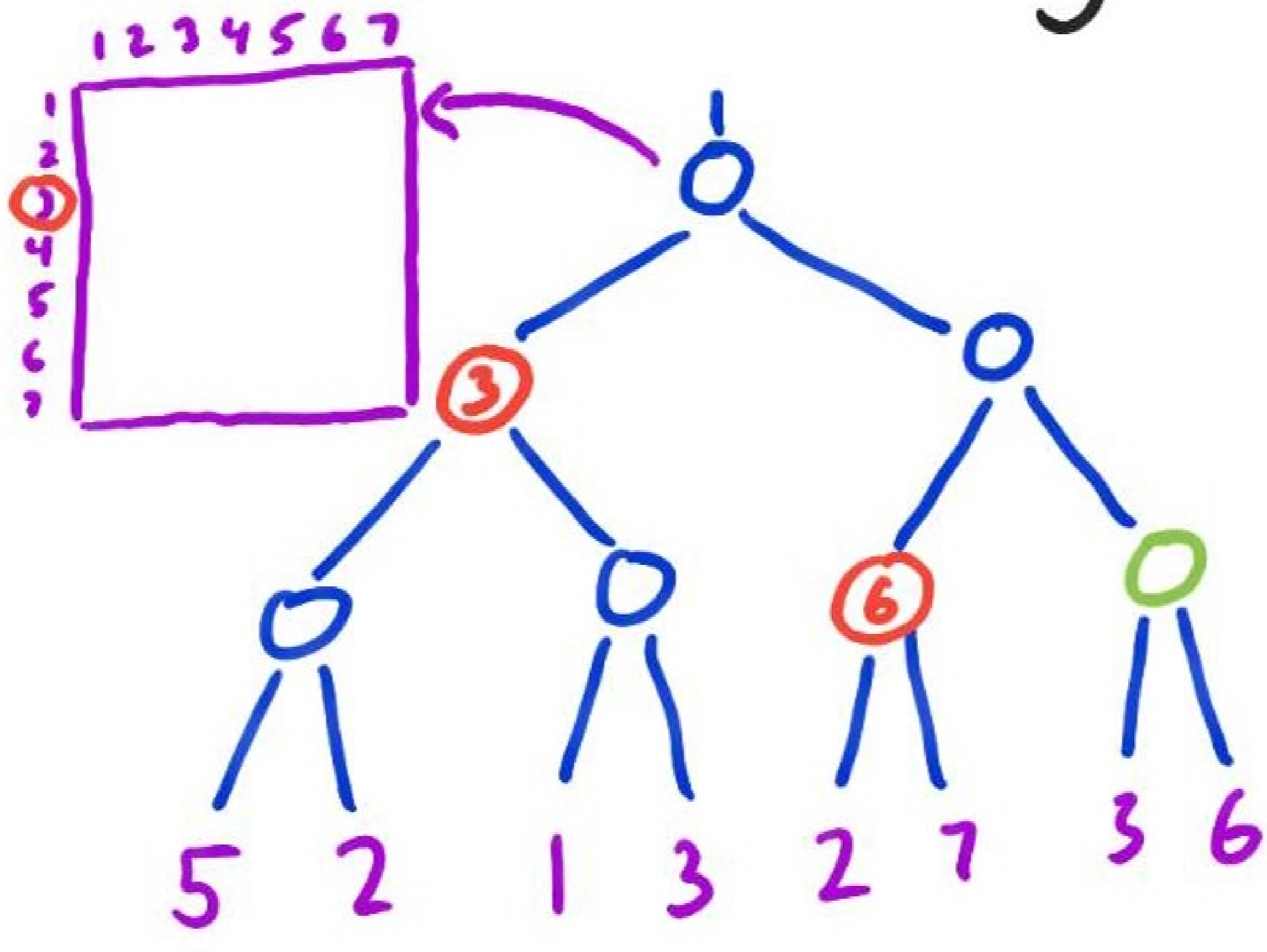
Pebbling



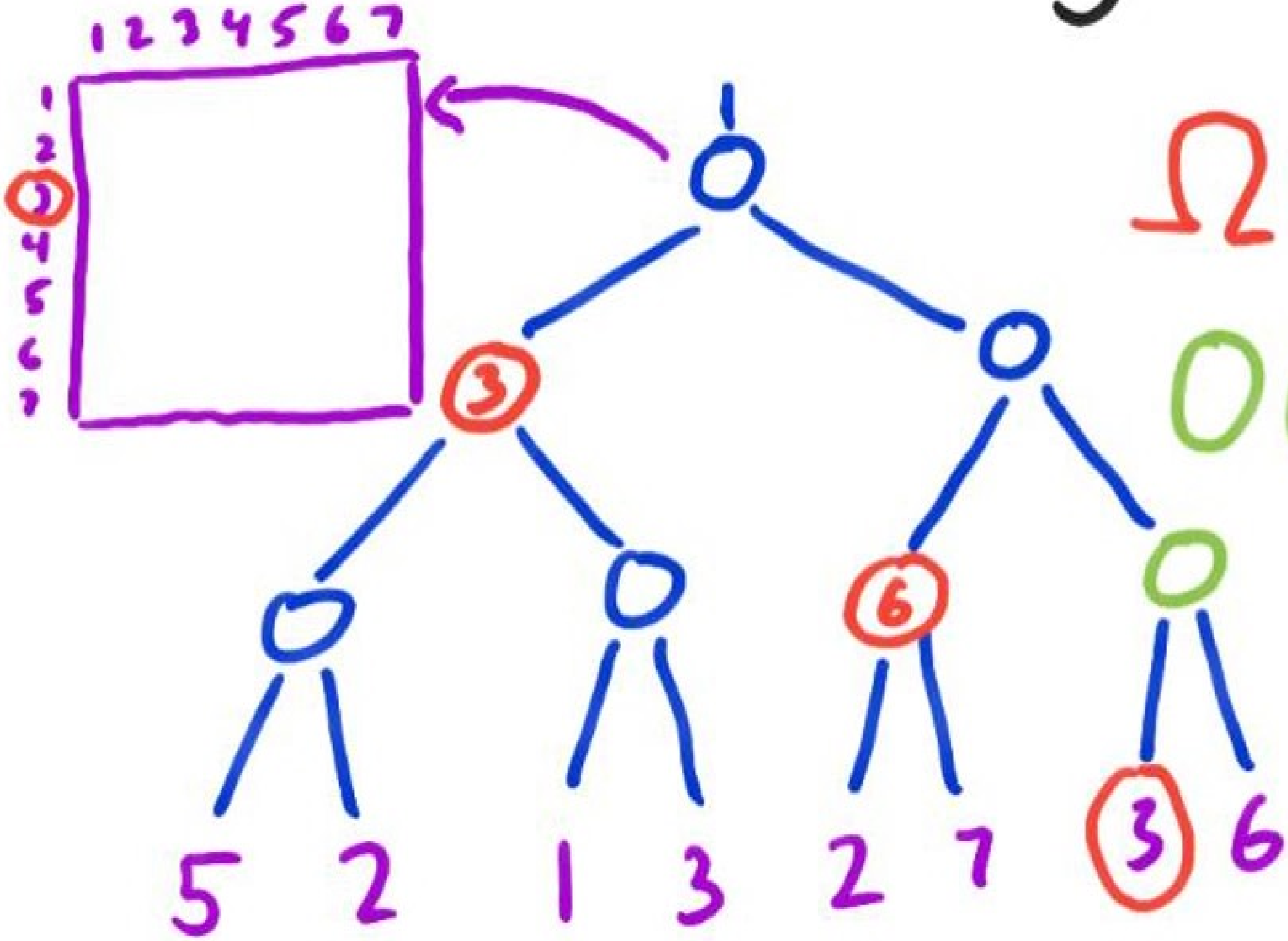
Pebbling



Pebbling



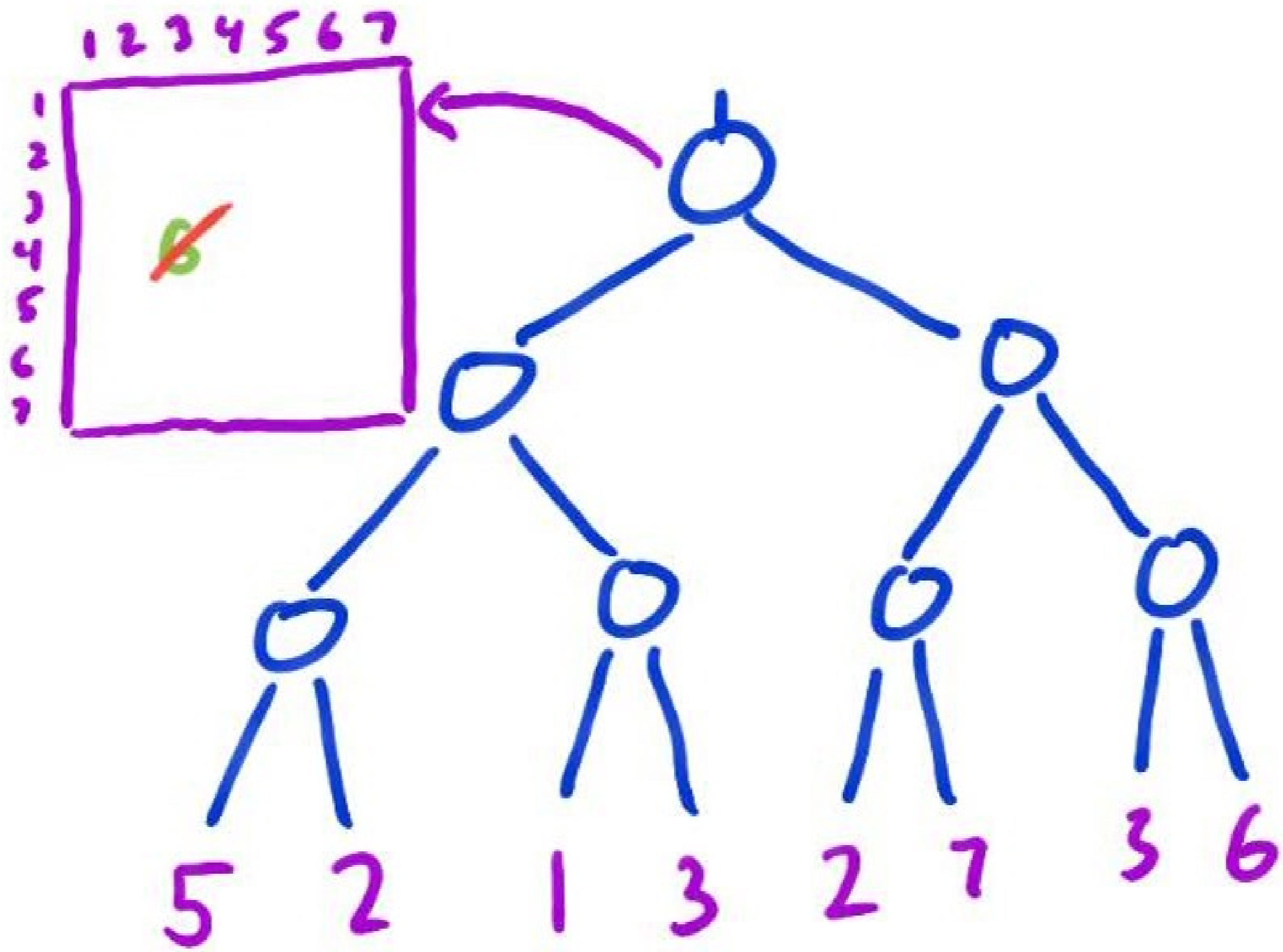
Pebbling



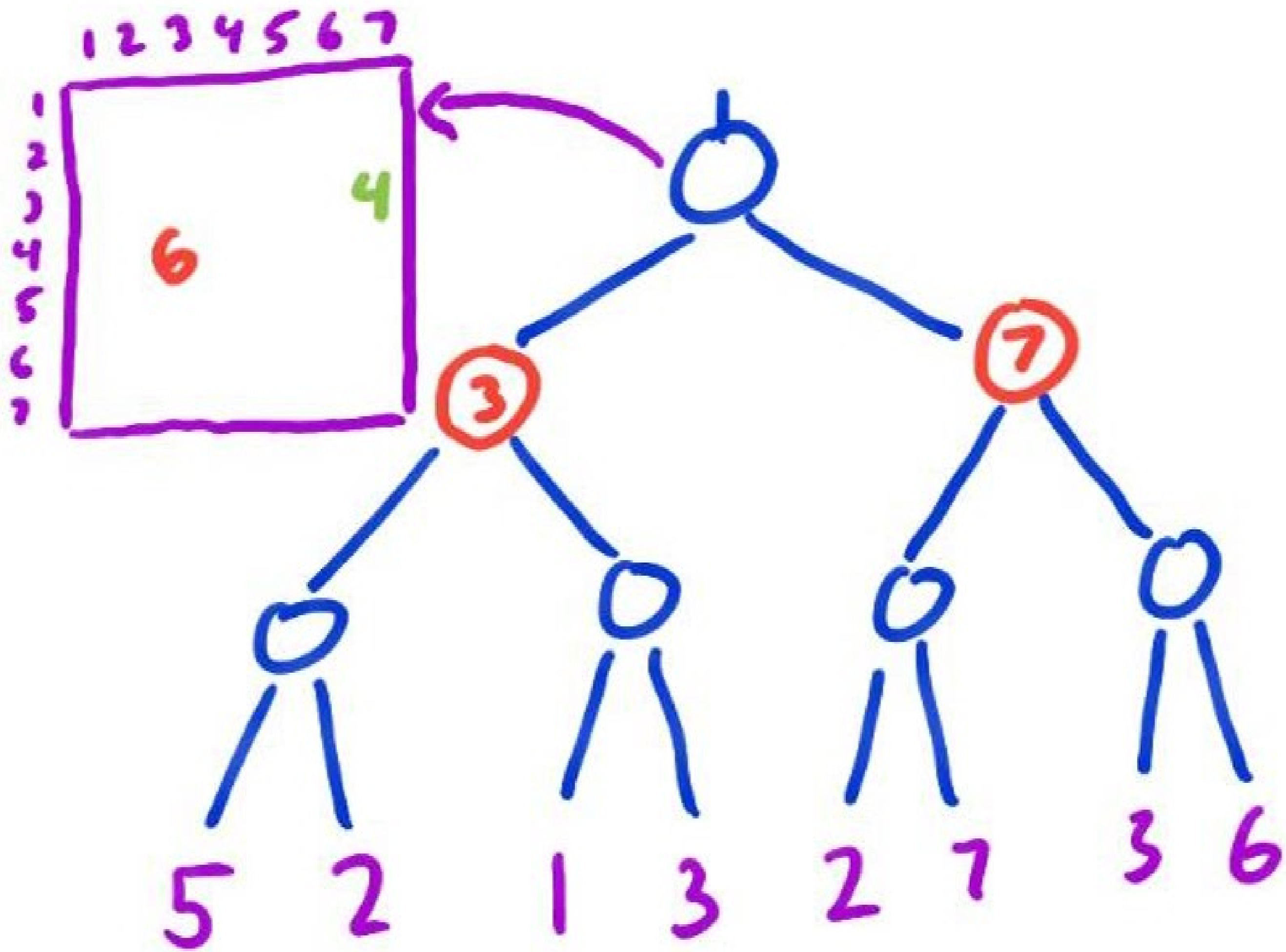
$\Omega(h \cdot \log k)$

$O(h + \log k)$

Restrictions (read-once)



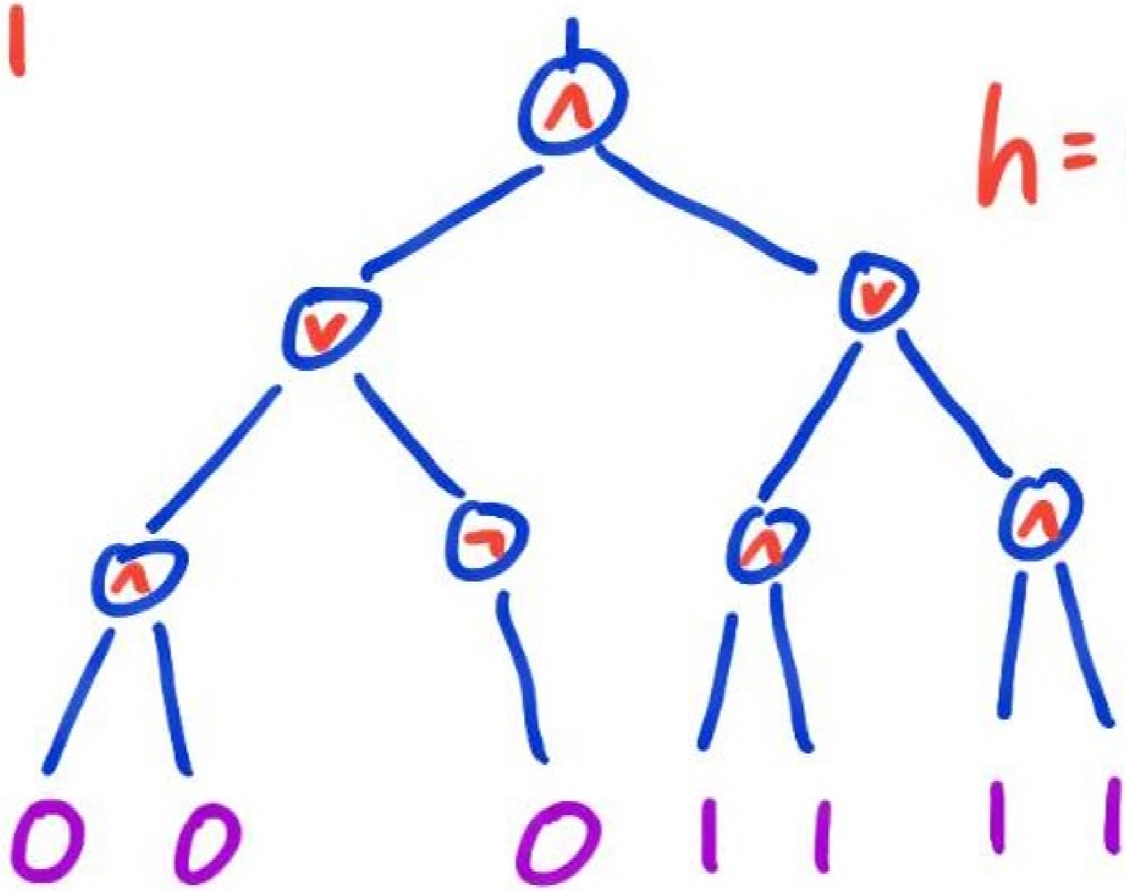
Restrictions (thrifty)



Barrington's Theorem

Barrington's Theorem

NC^1



$h = O(\log n)$

Barrington's Theorem

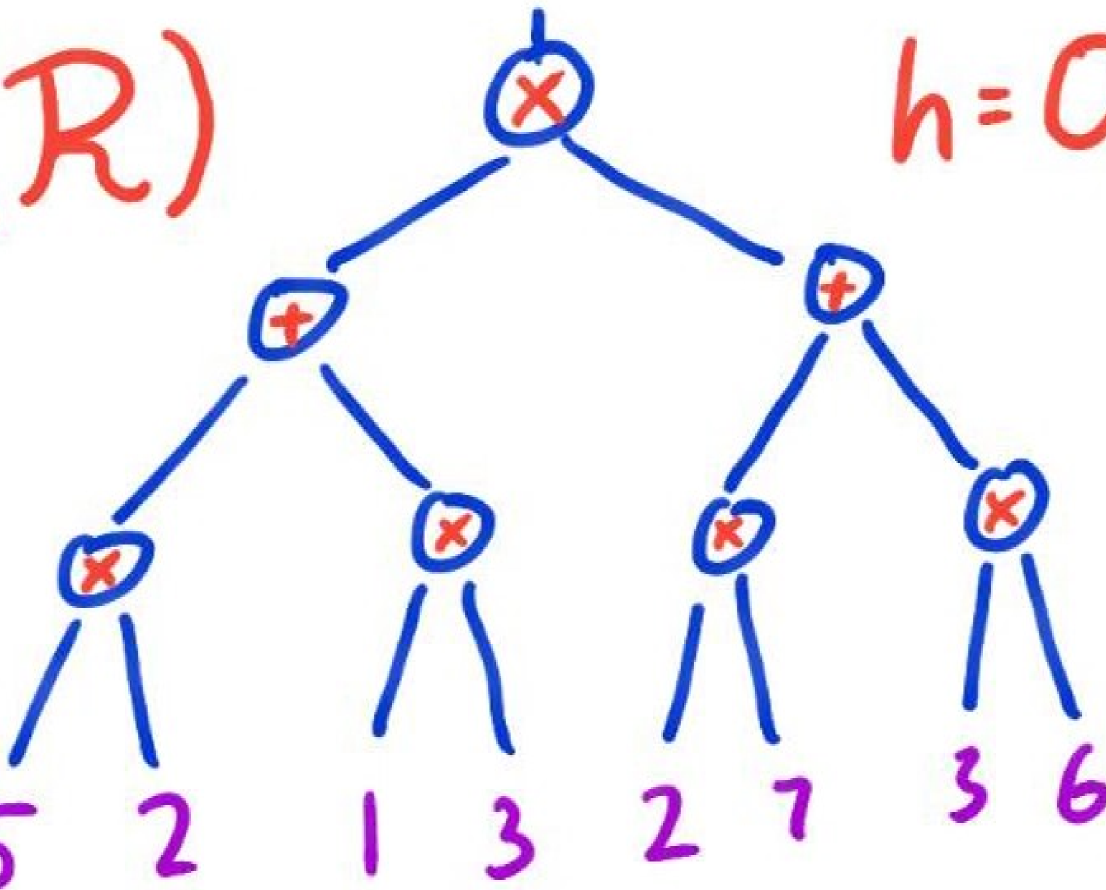
NC^1 can be computed by a layered width 5 poly-size deterministic branching program.

Ben-Or/Cleve '92

$\#NC^1(\mathcal{R})$

$h = O(\log n)$

$\mathcal{R} = \mathbb{F}_7$



Ben-Or/Cleve '92

Register program $P(1, 2 \ldots s)$

1. $R_1 \leftarrow R_1 + x_1 \cdot R_2$
2. $R_4 \leftarrow R_4 + R_2 \cdot R_3$
3. $R_1 \leftarrow R_1 + 8 \cdot x_3$

$\vdots$

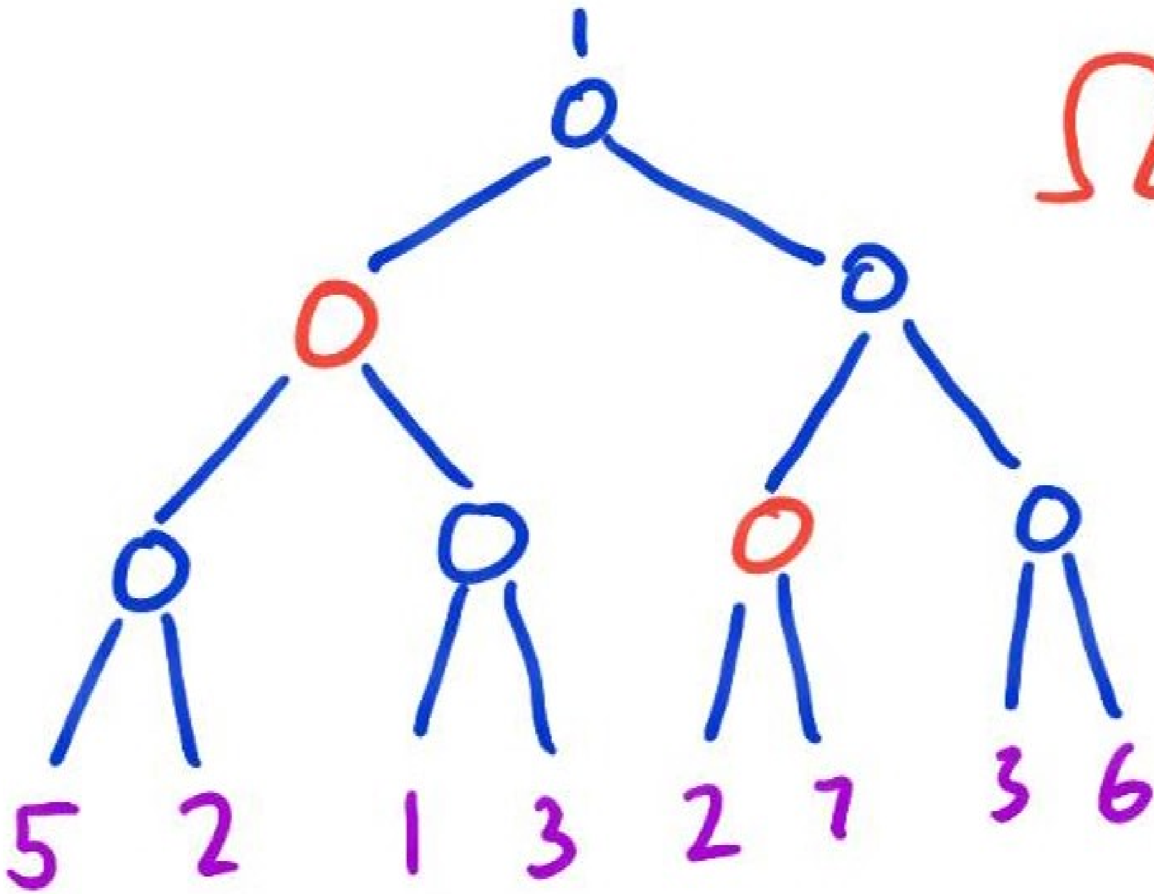
$$\underbrace{0}_{R_1} \quad \underbrace{1}_{R_2} \quad \cdots \quad \underbrace{0}_{R_s}$$

$$\boxed{\begin{array}{l} R_1 \leftarrow R_1 + f(\bar{x}) \cdot R_2 \\ R_j \leftarrow R_j \quad \forall j \neq 1 \end{array}}$$

Ben-Or/Cleve '92

$\#NC^1(R)$ can be computed by a poly-size register program over R with 3 registers.

Ben-Or/Cleve '92



$\Omega(\cancel{\log n})$

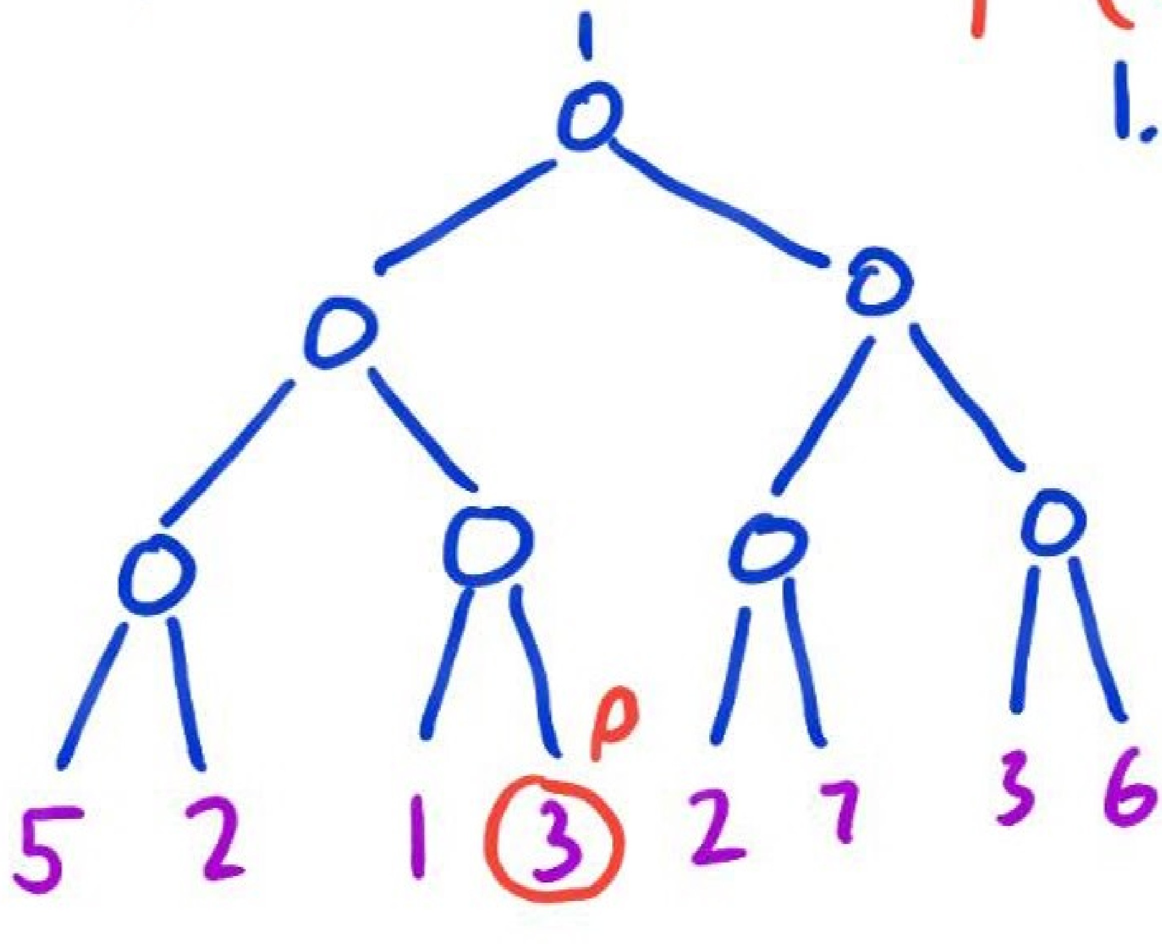
3 !

Ben-Or/Cleve '92

Proof:

Ben-Or/Cleve '92

Proof:

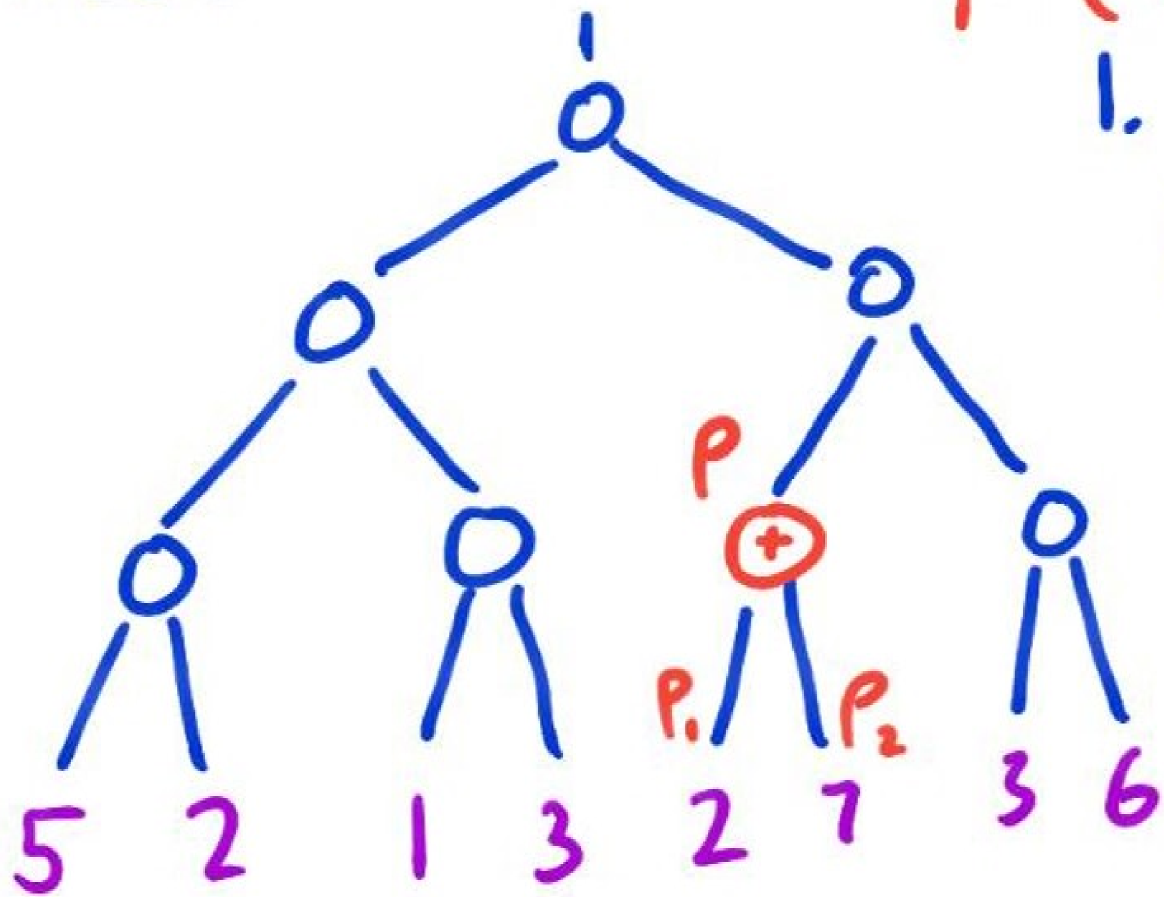


$P(1, 2, 3)$

1. $R_1 \leftarrow R_1 + 3 \cdot R_2$

Ben-Or/Cleve '92

Proof:



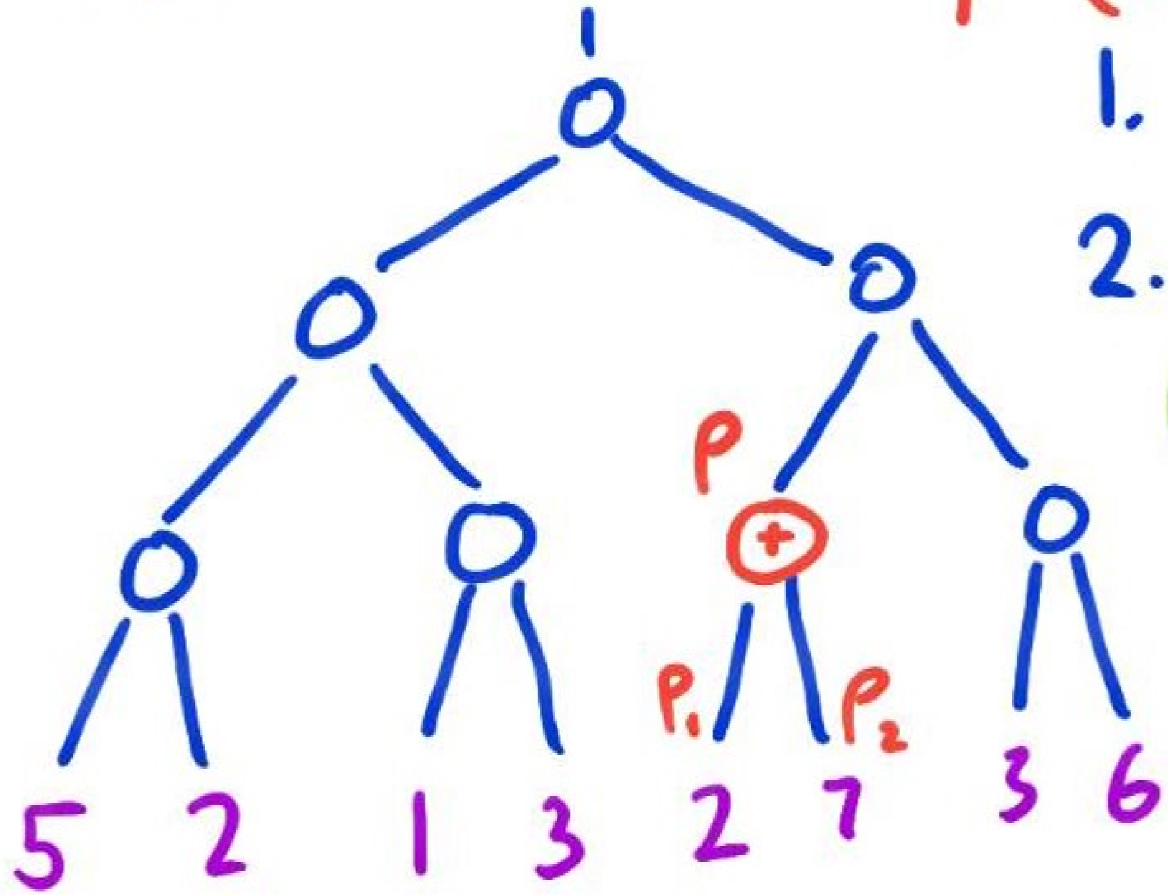
$P(1, 2, 3)$

1. $P_1(1, 2, 3)$

$R_1 \leftarrow R_1 + f_1 \cdot R_2$

Ben-Or/Cleve '92

Proof:



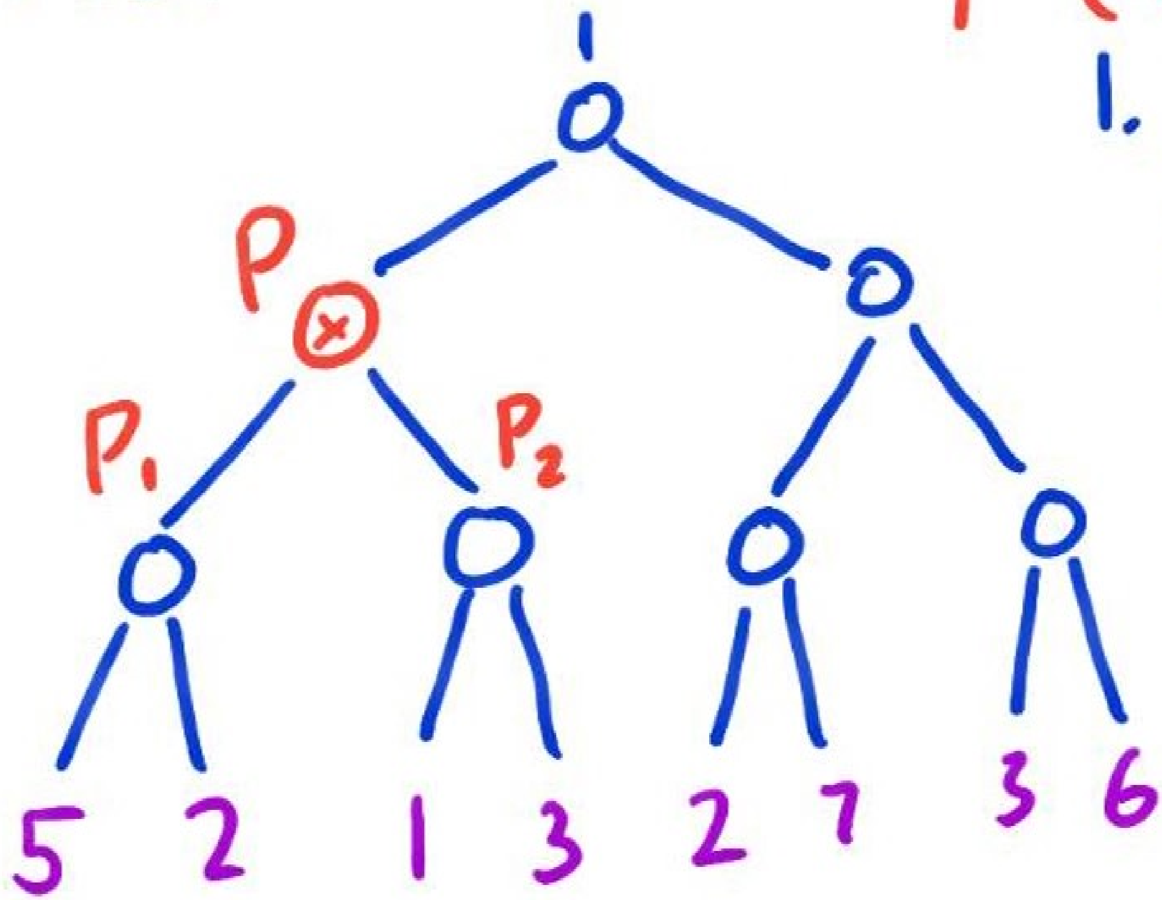
$P(1,2,3)$

1. $P_1(1, 2, 3)$
2. $P_2(1, 2, 3)$

$$R_1 \leftarrow (R_1 + f_1 \cdot R_2) + f_2 \cdot R_2$$

Ben-Or/Cleve '92

Proof:



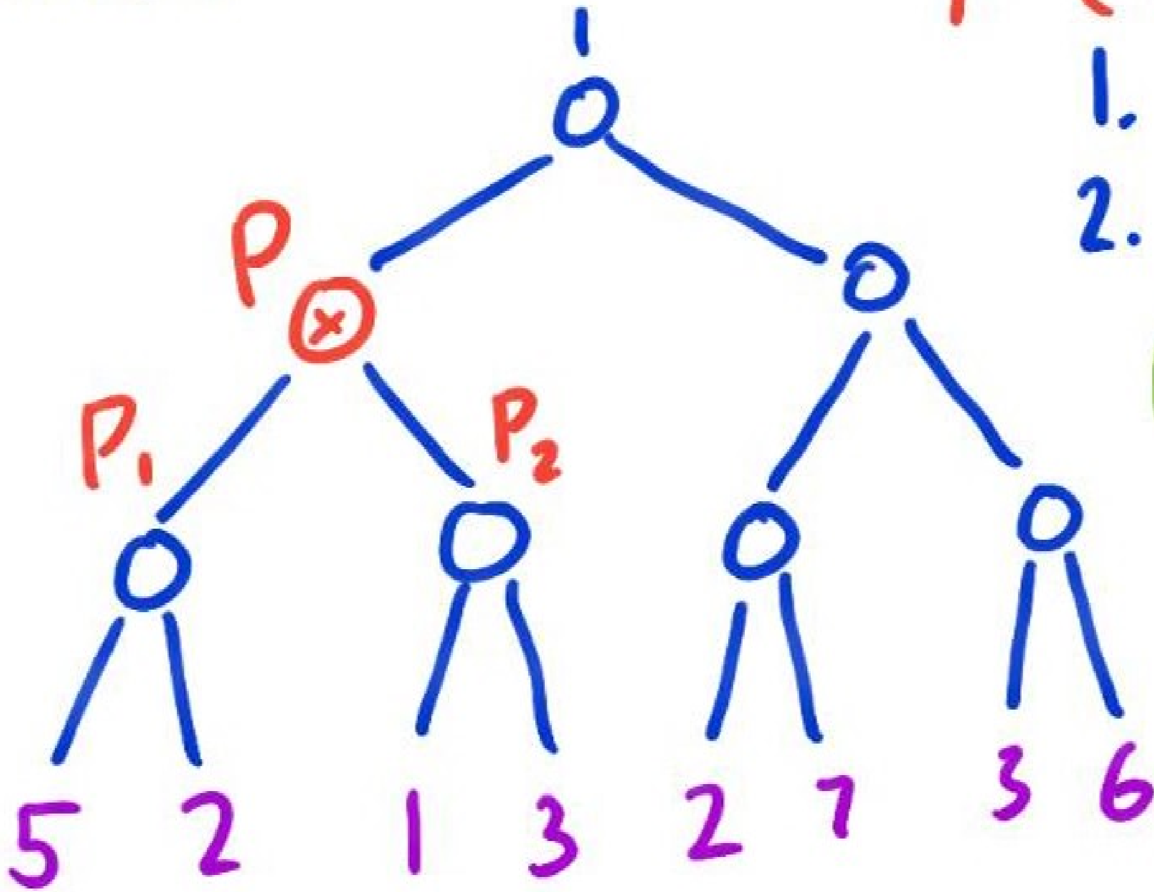
$P(1,2,3)$

1. $P_1^{-1}(1,3,2)$

$R_1 \leftarrow R_1 - f_1 \cdot R_3$

Ben-Or/Cleve '92

Proof:



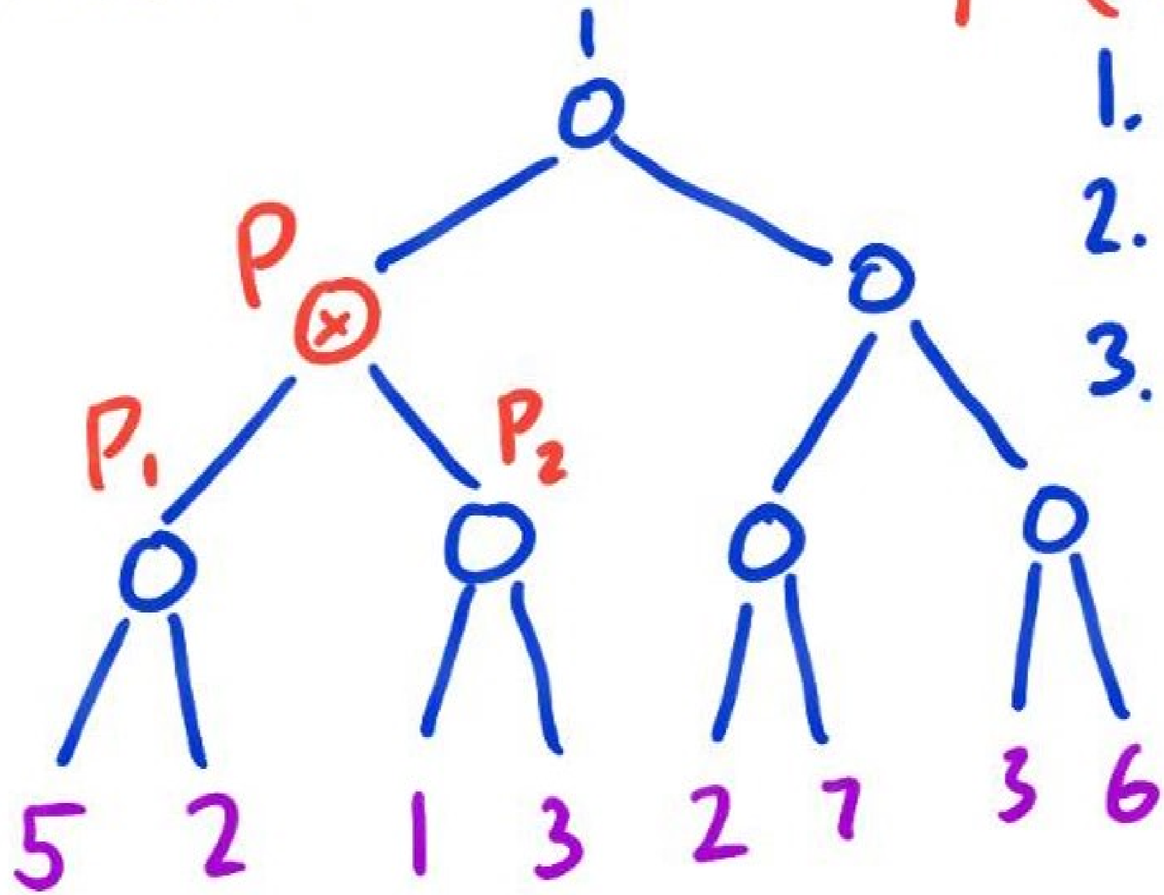
$P(1,2,3)$

1. $P_1^{-1}(1,3,2)$
2. $P_2(3,2,1)$

$R_3 \leftarrow R_3 + f_2 \cdot R_2$

Ben-Or/Cleve '92

Proof:



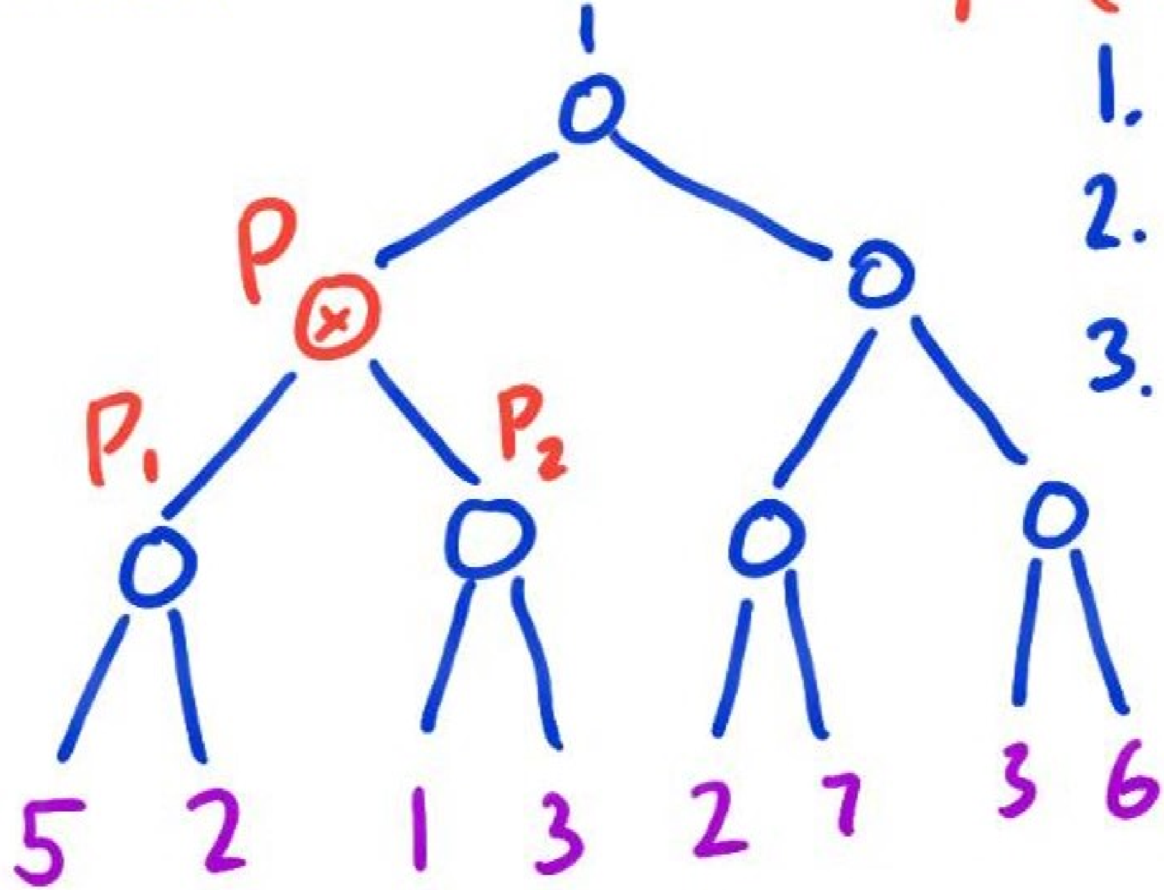
$P(1,2,3)$

1. $P_1^{-1}(1,3,2)$
2. $P_2(3,2,1)$
3. $P_1(1,3,2)$

$$R_1 \leftarrow (R_1 - f_1 \cdot R_3) + f_1 \cdot (R_3 + f_2 \cdot R_2)$$

Ben-Or/Cleve '92

Proof:



$P(1,2,3)$

1. $P_1^{-1}(1,3,2)$
2. $P_2(3,2,1)$
3. $P_1(1,3,2)$
4. $P_2^{-1}(3,2,1)$

$\blacksquare$

Catalytic + TEP

$\#NC^1(R)$ can be computed by a poly-size register program over R with 3 registers.

Catalytic + TEP

Given register programs P_1, P_2 where P_i transparently computes f_i, there exists a register program P which transparently computes $f_1 \cdot f_2$ using the <u>same</u> space and at most 4 program calls.

Catalytic + TEP

Given register programs P_1, P_2 where P_i transparently computes $\vec{f_i}$, there exists a register program P which transparently computes $\sum_{j,k} f_{1,j} \cdot f_{2,k}$ using the <u>same</u> space and at most 4 program calls.

Catalytic + TEP

Given register programs $P_1 \ldots P_d$ where P_i transparently computes f_i, there exists a register program P_d which transparently computes $\prod_{i=1}^{d} f_i$ using the <u>same</u> space and at most d^2 program calls.

Catalytic + TEP

Main Theorem [Buhrman+ '14]:

Catalytic Logspace contains TC^1 ($\supseteq NL$).

Catalytic + TEP

<u>Main Theorem</u> [Cook-Mertz '19]:

$TEP_{h,k}$ can be simulated by a register program with $\left(\frac{k}{\varepsilon h}+1\right)^{2h}$ instructions and $\left(\frac{k}{\varepsilon h}+1\right)^{3\varepsilon h}$ boolean registers.

Catalytic + TEP

$$(2+3\varepsilon)\, h \cdot \log\left(\frac{k}{\varepsilon h}+1\right) \text{ vs } h \cdot \log k$$

Catalytic + TEP

$(2+3\varepsilon)\, h \cdot \log\left(\frac{k}{\varepsilon h}+1\right)$ vs $h \cdot \log k$

- better for $h = \omega\left(k^{1/2+\varepsilon}\right)$

Catalytic + TEP

$(2+3\varepsilon)\, h \cdot \log\left(\frac{k}{\varepsilon h}+1\right)$ vs $h \cdot \log k$

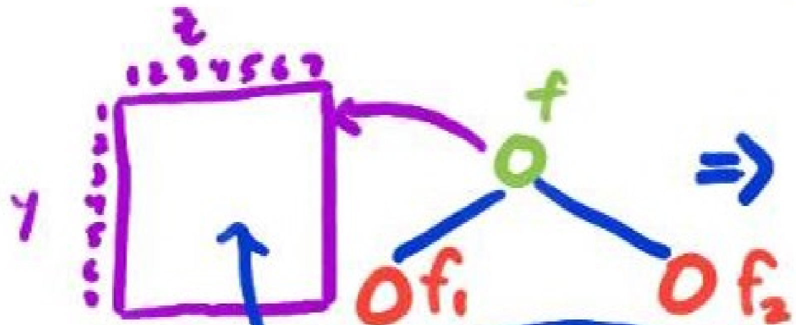
- better for $h = \omega\left(k^{1/2+\varepsilon}\right)$
- not read-once

1. $P_1^{-1}(1,3,2)$
2. $P_2(3,2,1)$
3. $P_1(1,3,2)$
4. $P_2^{-1}(3,2,1)$

Catalytic + TEP

$(2+3\varepsilon)\, h \cdot \log\left(\frac{k}{\varepsilon h}+1\right)$ vs $h \cdot \log k$

- better for $h = \omega\left(k^{1/2+\varepsilon}\right)$
- not read-once or thrifty



$\Rightarrow [f = x] = \sum_{y,z} \underbrace{[f(y,z) = x]}\,[f_1 = y]\,[f_2 = z]$

Open problems

Open problems

1) P computes $\sum_j \prod_i f_{i,j}$ using $t(n) < 2^n$ program calls

$$\mathrm{poly}(n) \Rightarrow TEP \in (\log k)^{O(k)}$$
$$O(1) \Rightarrow TEP \in L$$

Open problems

1) P computes $\sum_j \prod_i f_{i,j}$ using $t(n) < 2^n$ program calls

2) use lemmas to simulate
   - AC^1 in non-catalytic L

Open problems

1) P computes $\sum_j \prod_i f_{i,j}$ using $t(n) < 2^n$ program calls

2) use lemmas to simulate
   - AC^1 in non-catalytic L
   - TC^1 in VP