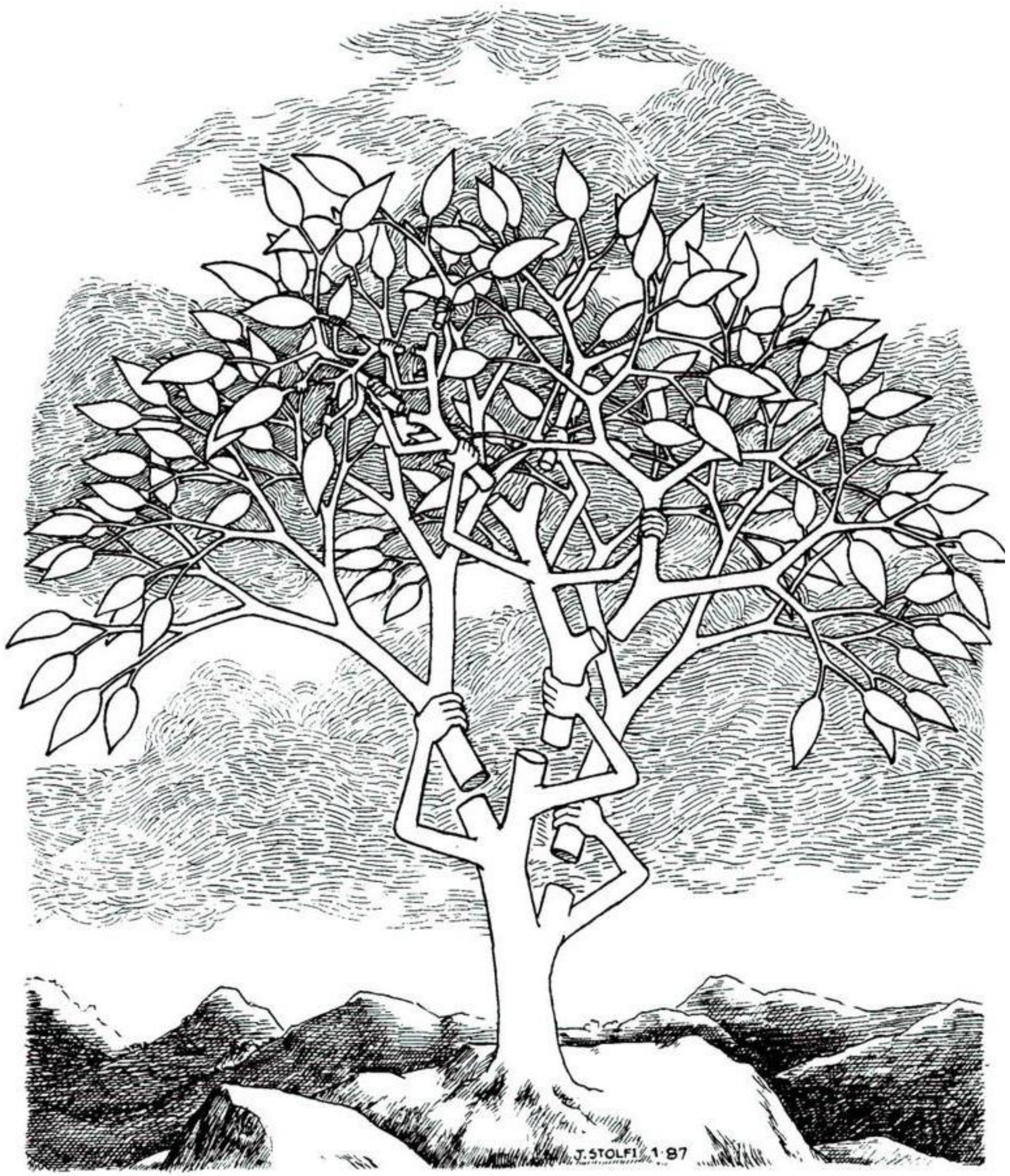
Dynamically Optimal Self-Adjusting Single-Source Tree Networks



A Self-Adjusting Search Tree By Jorge Stolfi (1987)

Chen Avin

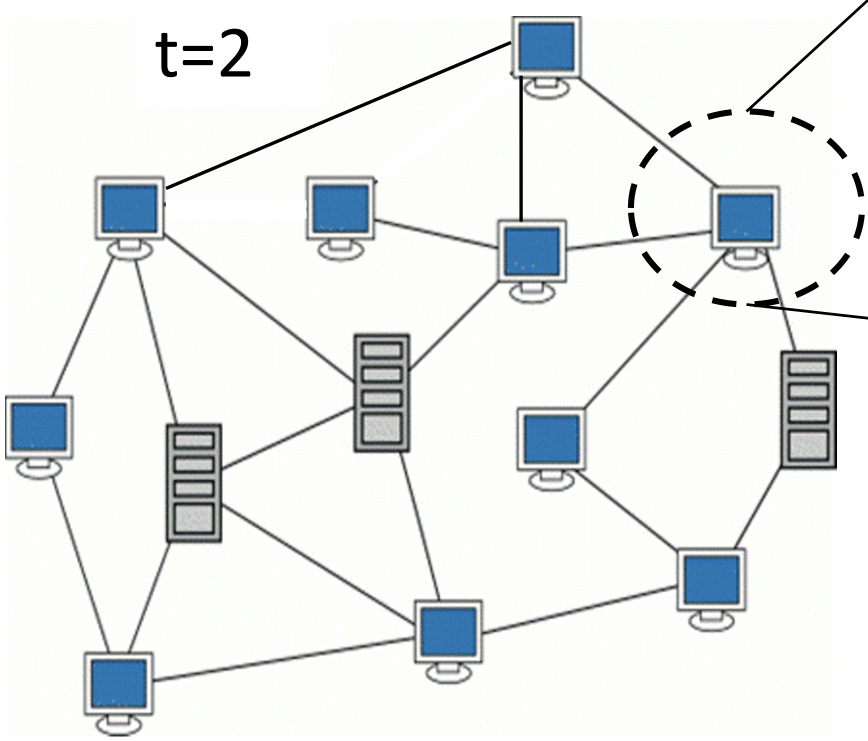
Ben-Gurion University of the Negev

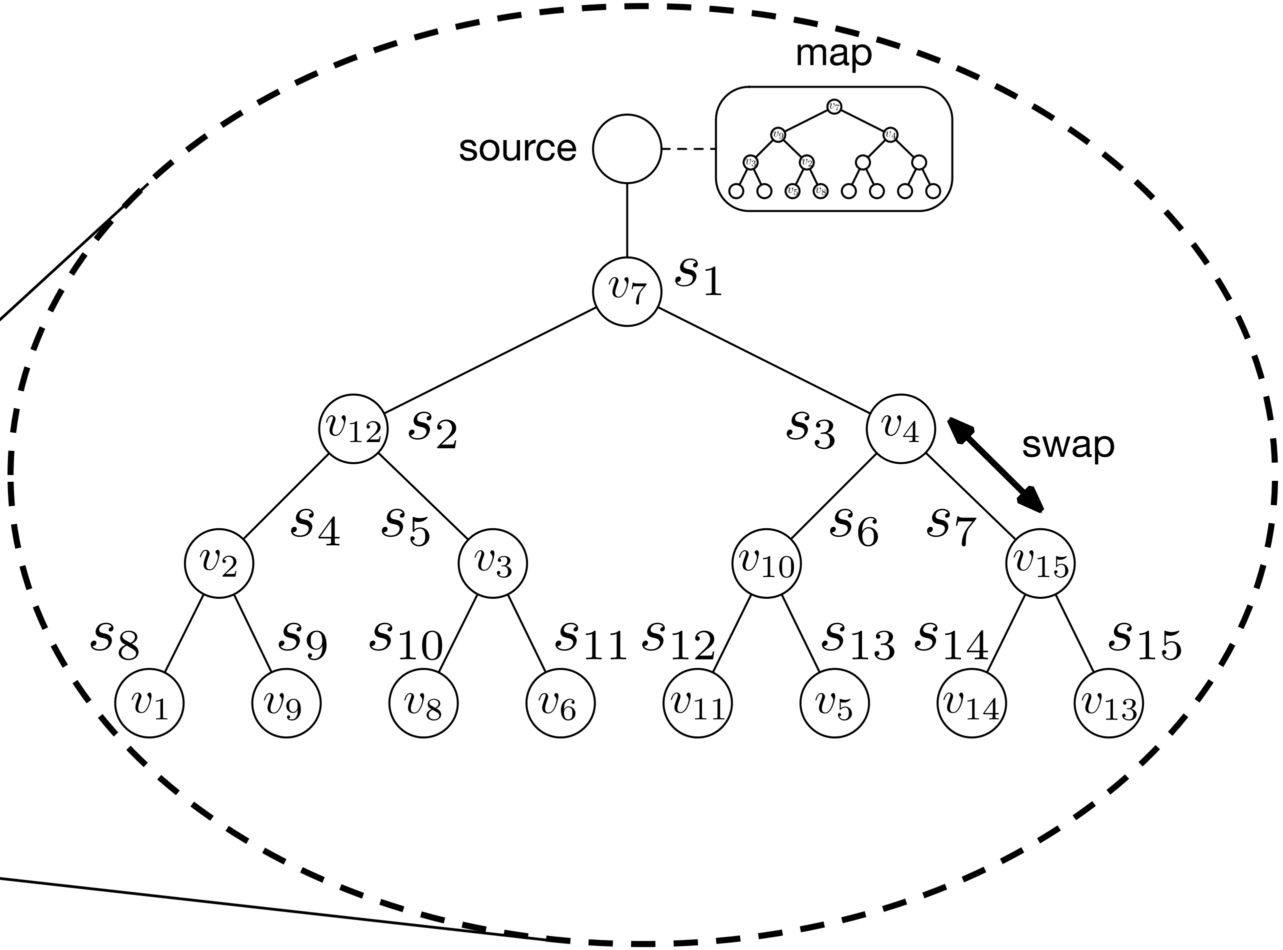
Joint work with Kaushik Mondal and Stefan Schmid

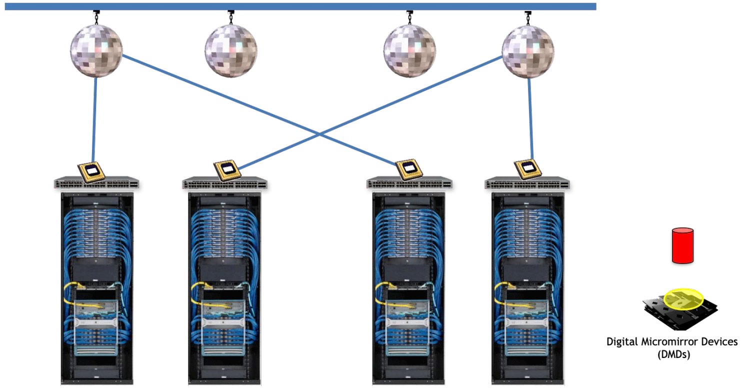
Latin 2020
January 5th - 8th
São Paulo, Brazil

Motivation

- Self Adusting Networks







e.g.,
ProjecToR @ SIGCOMM 2016
(Microsoft)

Model (self-adjusting list)

- Dynamic list update [ST85]

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

- Sequnnce σ
- Unit operation: **swap**
- Move-To-Front (MTF)
- Most-Recently-Used (MRU) property
- Working-Set property
- MTF is dynamicly optimal !

$$\mathrm{cost}(\mathrm{ALG}(\sigma))$$

$$= \sum_{t=1}^{m} cost\left(ALG\left(\sigma^{(t)}\right)\right)$$

$$= \sum_{t=1}^{m} \mathrm{acc{-}cost}^{(t)} + \mathrm{adj{-}cost}^{(t)}$$

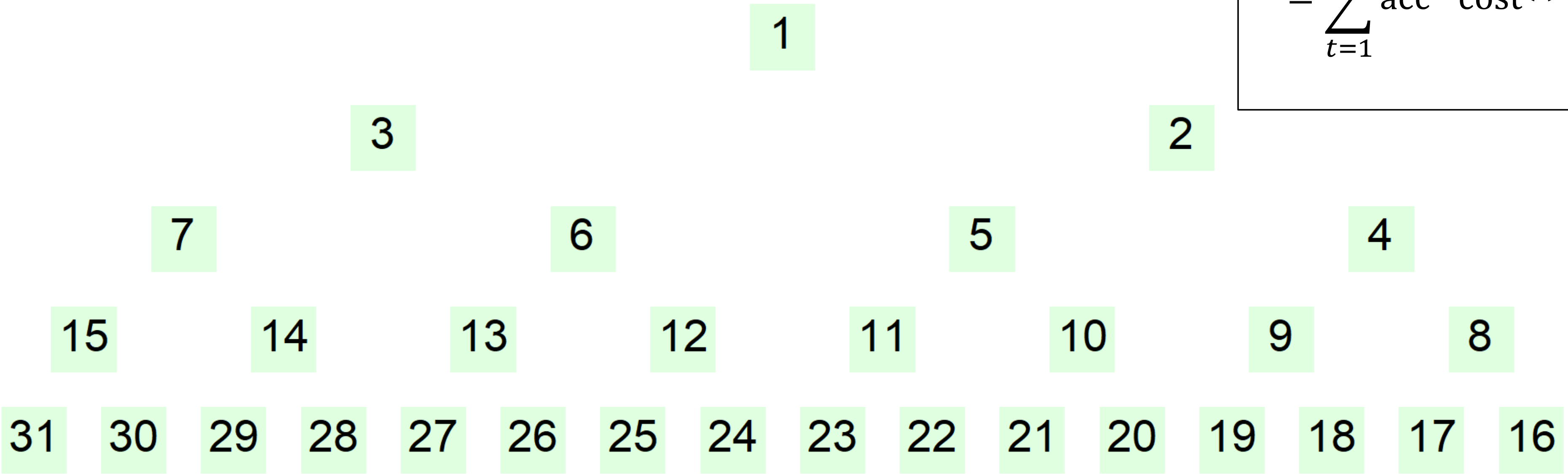
Let see an example

List-update

Model (self-adjusting complete tree)

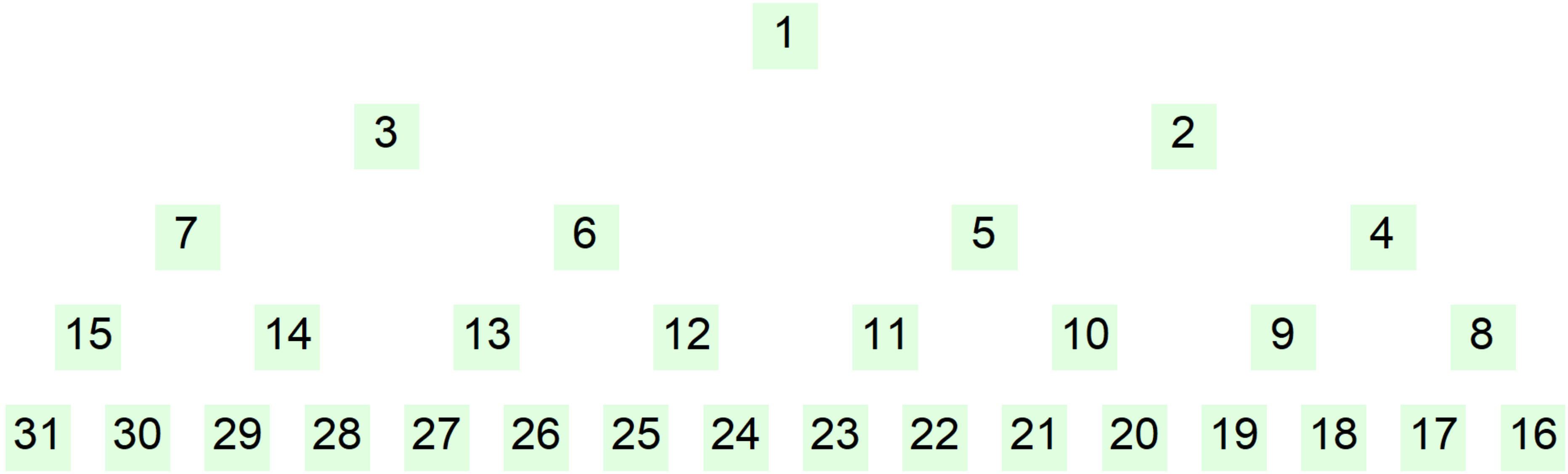
- Unit operation: **swap**
- Simple MTF doesn't work
- Where to push down?

$$\begin{aligned}&\mathrm{cost}(\mathrm{ALG}(\sigma))\\ &= \sum_{t=1}^{m} cost\left(ALG(\sigma^{(t)})\right)\\ &= \sum_{t=1}^{m} \mathrm{acc{-}cost}^{(t)} + \mathrm{adj{-}cost}^{(t)}\end{aligned}$$



Lower bound: MRU Tree

- Working set bound: $\text{cost}(\text{ALG}(\sigma)) \geq \text{WS}(\sigma) = \sum_{t=1}^{m} \log(\sigma^{(t)}).\text{rank}$
- MRU tree is dynamicly optimal
- Proof by potential function

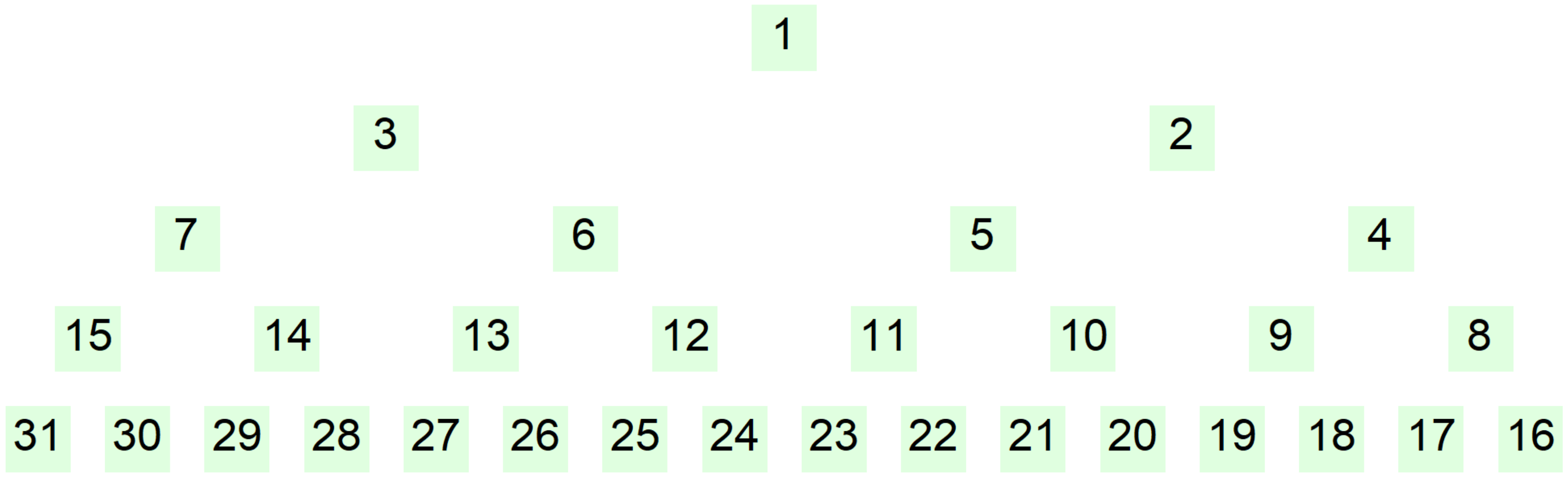


Let see an example

MRU tree

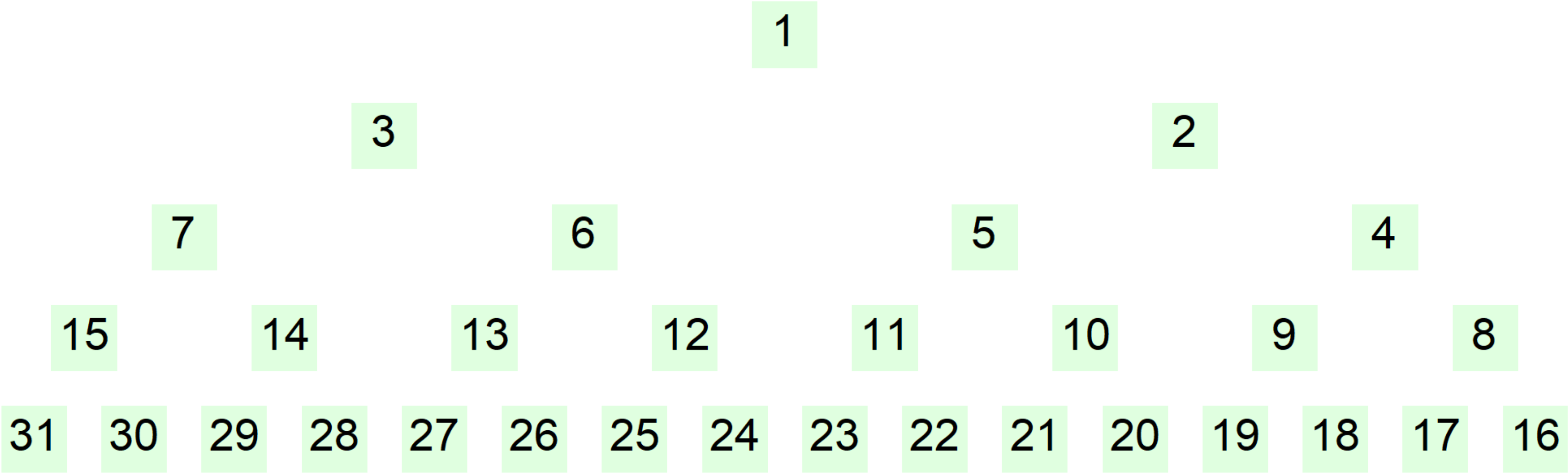
Alternative 1: MRU Tree via MAX-PUSH

- MAX-PUSH is too expensive



Alternative 2: RANDOM-PUSH

- Random walk to level k + MTF



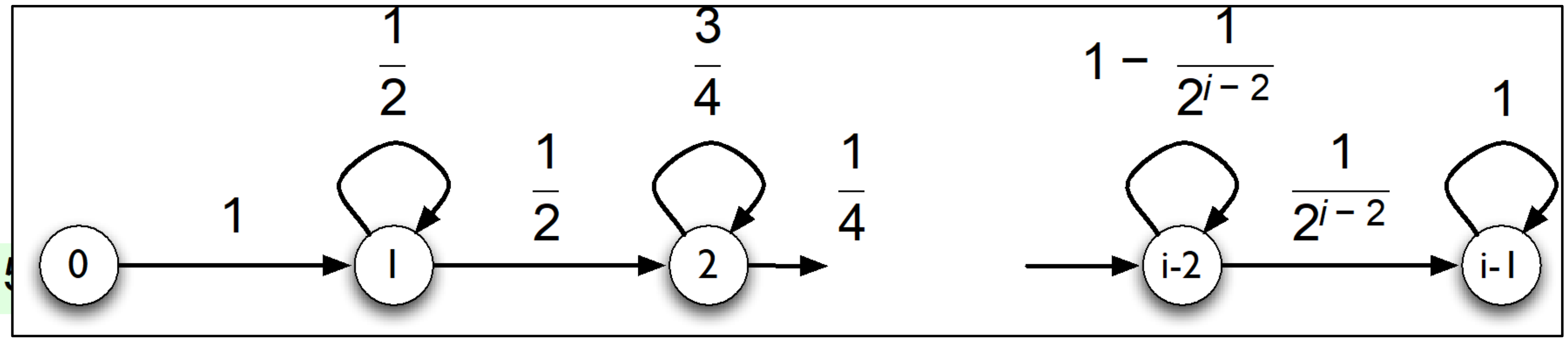
Let see an example

RANDOM-PUSH

Alternative 2: RANDOM-PUSH

- Theorem: The RANDOM PUSH algorithm is dynamically optimal on expectation.

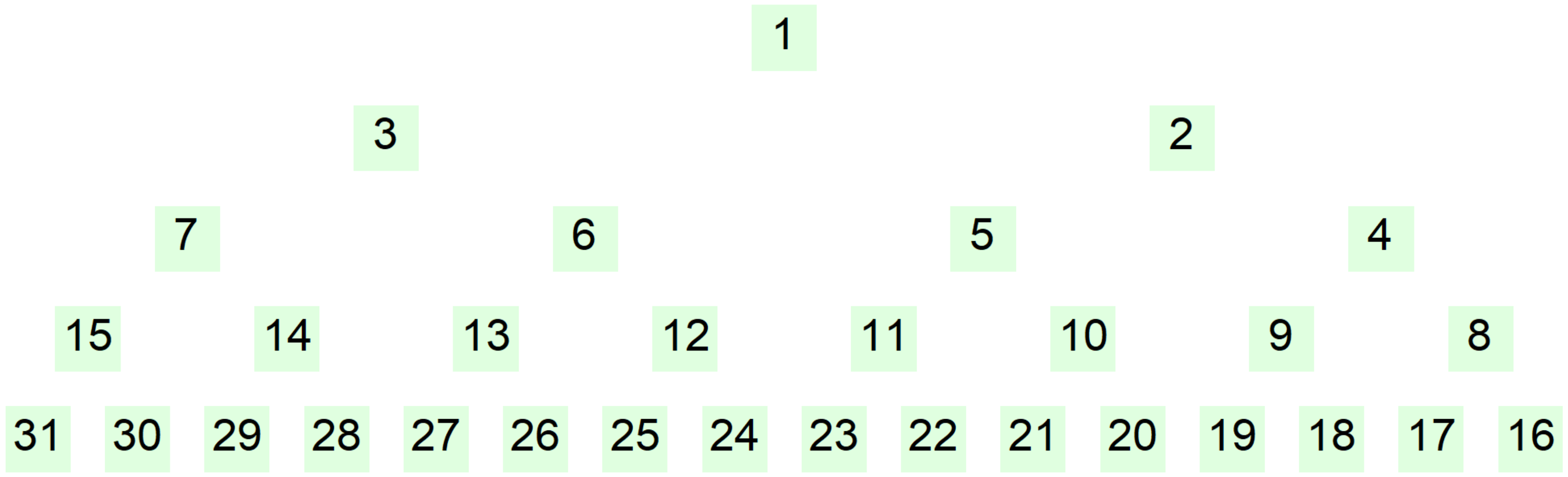
1

15 

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

Alternative 3: MOVE-HALF

- Swap with the maximum rank in level $\left\lfloor \frac{k}{2} \right\rfloor$
- Does not maintain the MRU property!

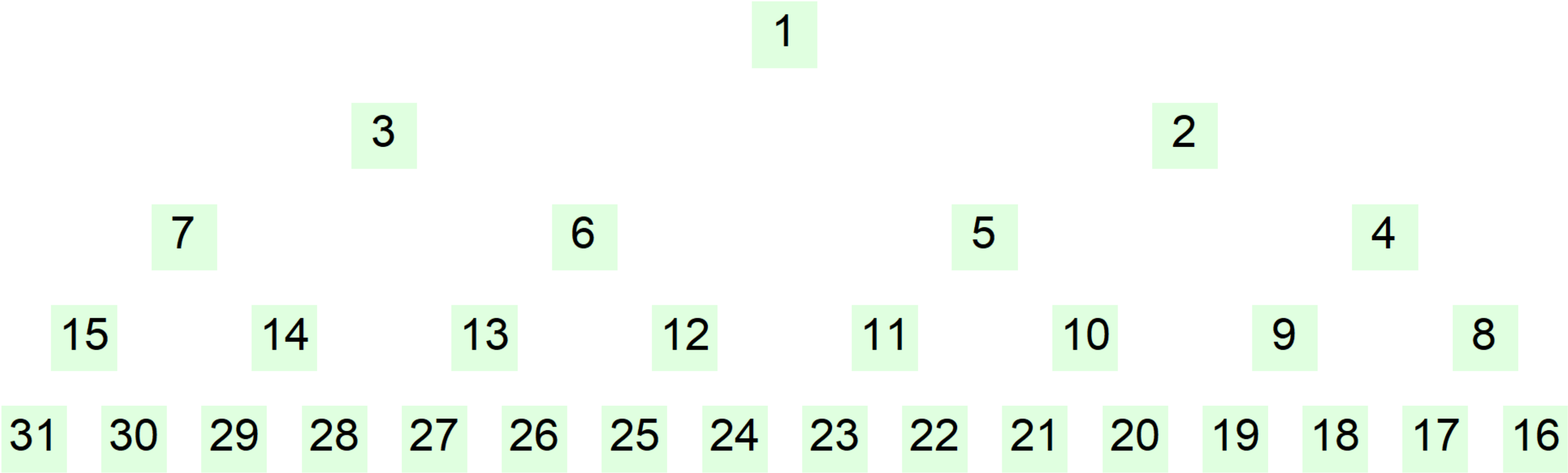


Let see an example

MOVE-HALF

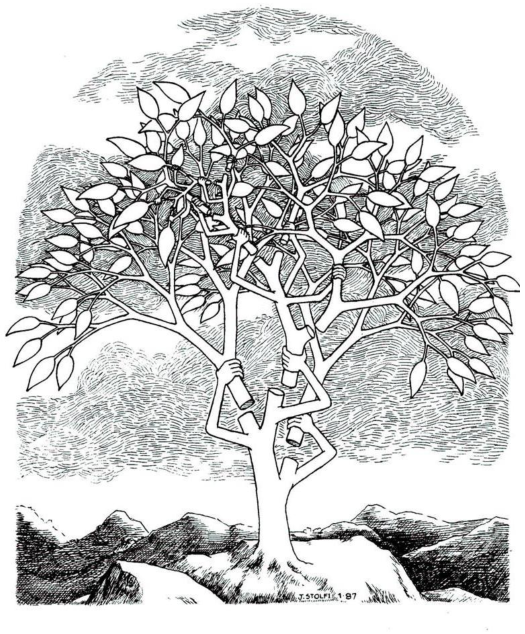
Alternative 3: MOVE-HALF

- MOVE-HALF is access-optimal (similar cost to MRU)!
- MOVE-HALF is dynamic optimal (since adjustment cost is $O(k)$)



Related Work

- Linked List
- Binary Search Trees (e.g., Splay trees)
- Unordered Trees
- Skip Lists and B-Trees
- Self-Adjusting Networks



Future Work

- Self-Adjusting ego-trees as bulding block for self-adjusting netwroks
- More powerfull adversary
- Routing
- Switch based models (e.g., the ToR-matching-ToR model)

Thank you!

Feel free to contact me: avin@cse.bgu.ac.il