Randomized Data Structures and Algorithms

John Augustine (IIT Madras)

Krishna Palem (RICE University)

Source:
These slides are based on source material (including notation, concepts, and presentation approach) https://www.ics.uci.edu/~pattis/ICS-23/lectures/notes/Skip%20Lists.pdf and http://www.cc.gatech.edu/~vigoda/7530-Spring10/Kargers-MinCut.pdf.

Overview

- Skip List: A Randomized Dictionary
  - Overall Structure & Operations
  - Analysis: number of levels, search time, space.
- Karger's Mincut Algorithm
  - Problem Definition
  - Algorithm
  - Analysis

Skip List

Its overall structure and operations.

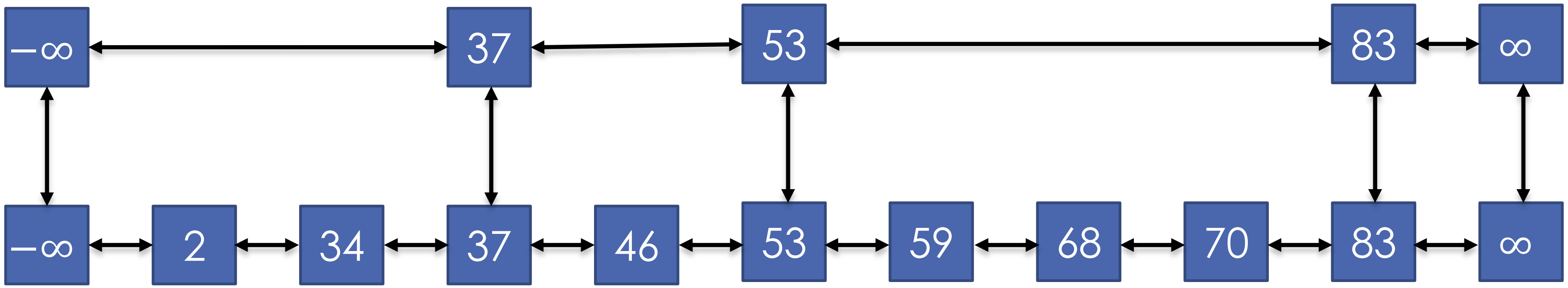
Recollecting the Dictionary ADT

- Stores $\langle \text{key}, \text{value} \rangle$ pairs.
- Operations: Insertion, Deletion, Search.
- Examples: (Balanced) Binary Search Trees.
- Typically, $O(n)$ space and $O(\log n)$ time per operation.
- Skip list achieves these bounds on expectation and with high probability (WHP), i.e., probability at least $1 - \frac{1}{n^c}$ for some fixed $c > 0$.

Intuition

- A linked list requires $\Omega(n)$ time for search
  - Why? Because the search procedure works like a slow train that stops at every station.
- How to speedup search time?
  - As a first thought, add an express lane.
  - With probability $\frac{1}{\sqrt{n}}$, promote each node to the express lane.

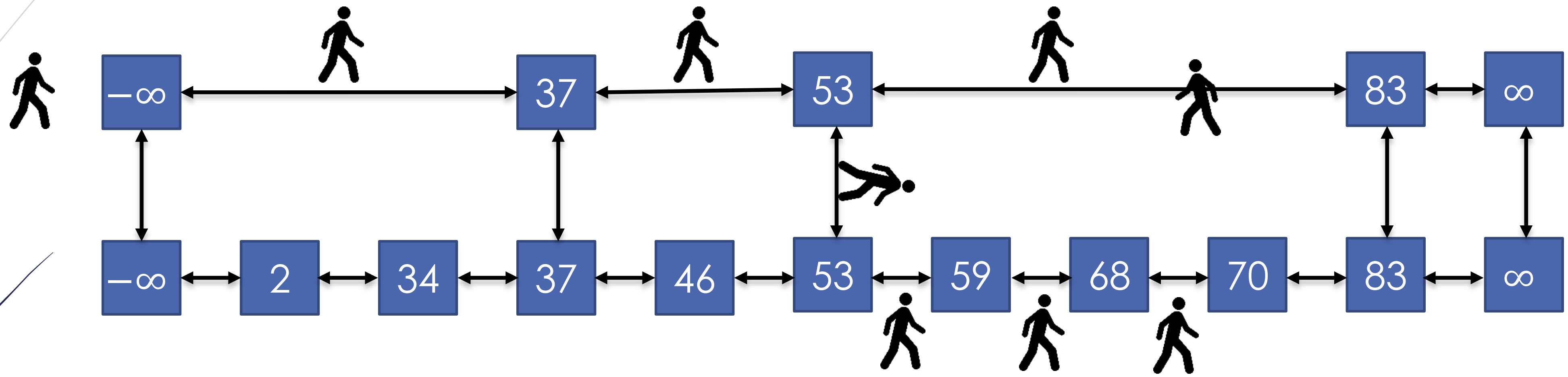
Intuition



- Sentinel nodes (with $-\infty$ and ∞ values) are present in all lanes or levels.
- There will now be $\mathbb{E}[\sqrt{n}]$ elements in the express lanes
- The number of elements in slow lane between two express lane elements is also $\sim\mathbb{E}[\sqrt{n}]$.
- Thus, expected times for each operation is $O(\sqrt{n})$. (Verify!)

Intuition

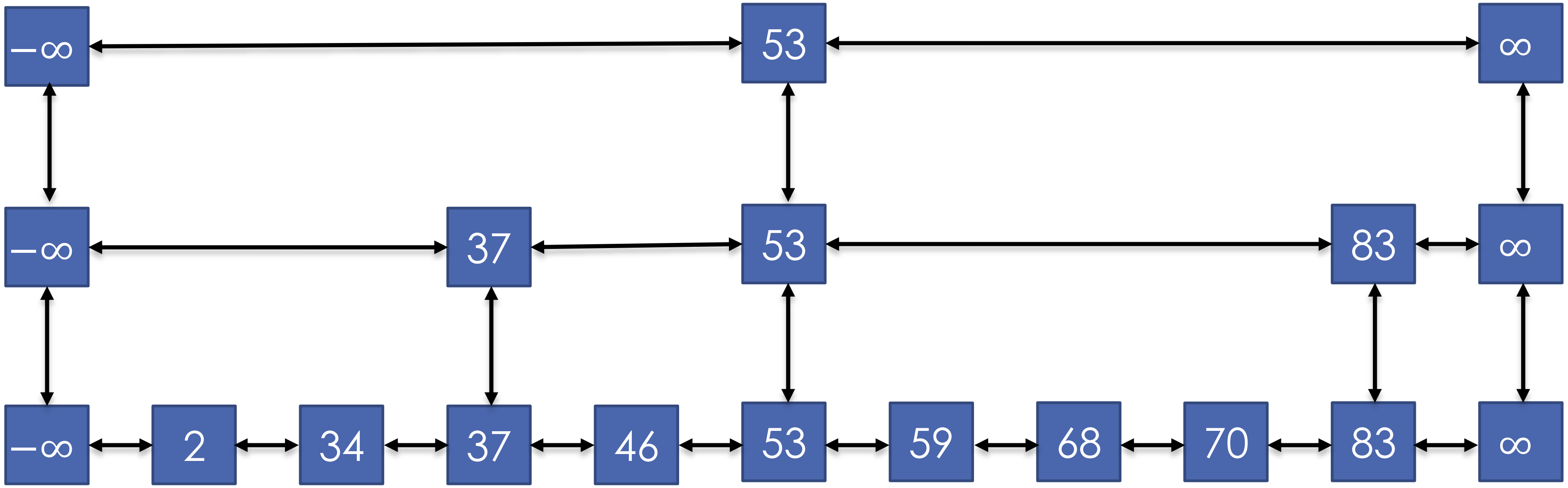
Searching for key value 70



- Sentinel nodes (with $-\infty$ and ∞ values) are present in all lanes or levels.
- There will now be $\mathbb{E}[\sqrt{n}]$ elements in the express lanes
- The number of elements in slow lane between two express lane elements is also $\sim\mathbb{E}[\sqrt{n}]$.
- Thus, expected times for each operation is $O(\sqrt{n})$. (Verify!)

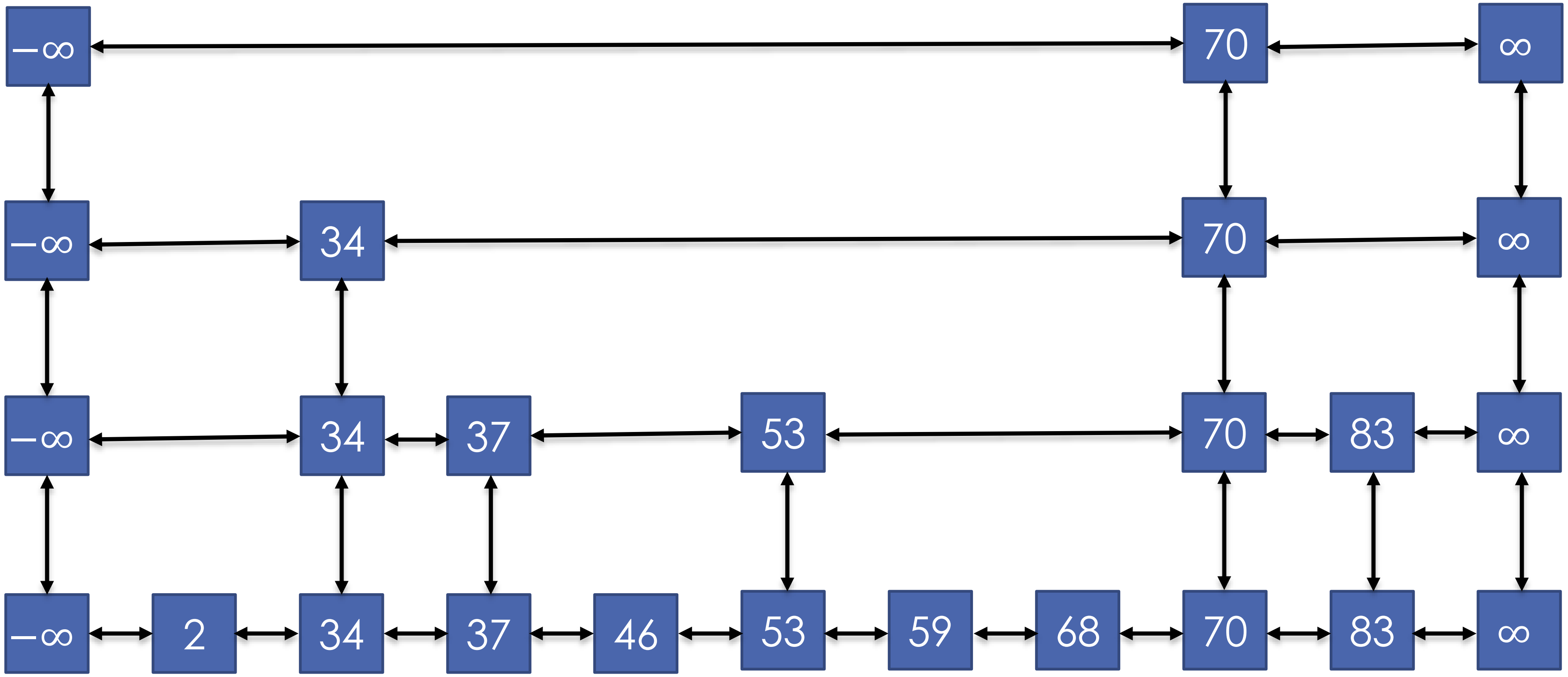
Extending the Intuition

- If one express lane helps, what about more?
  - Three? Four? … 100? … $\log \log n$? … $\log n$? … $\sqrt{n}$? …

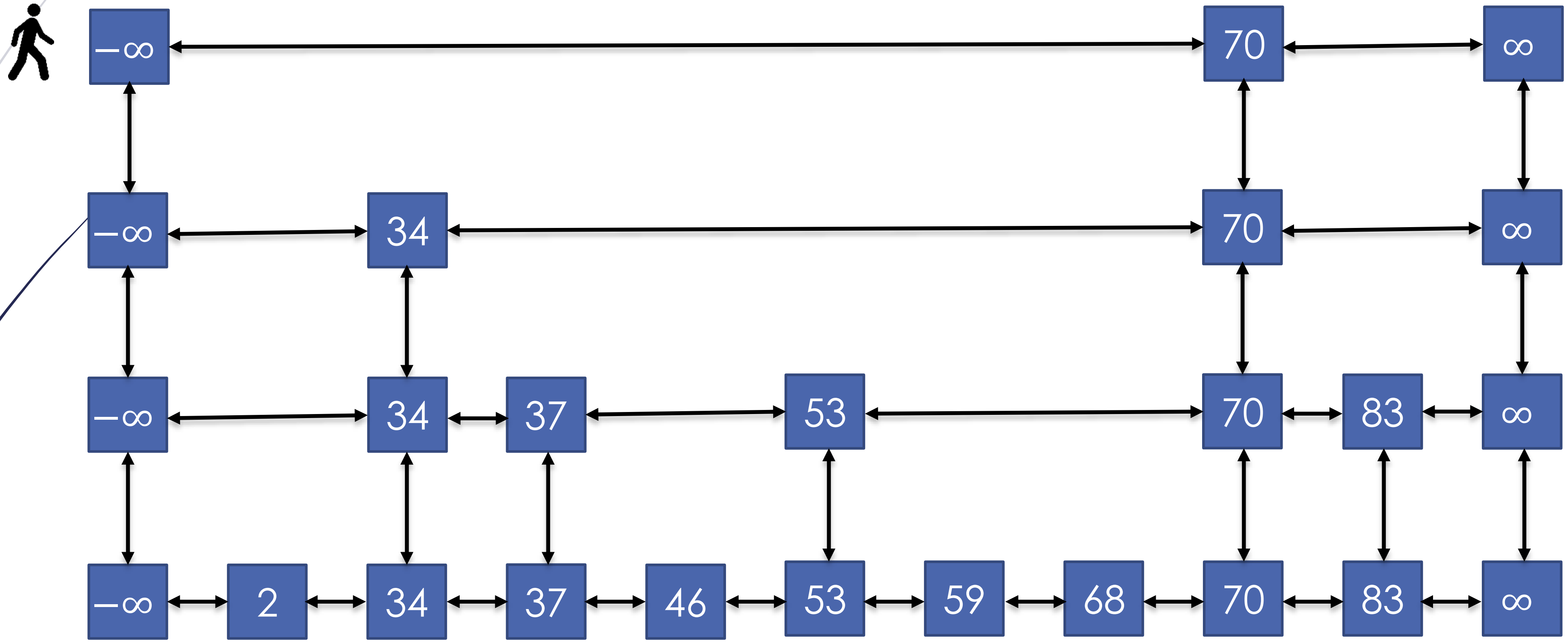


The Skip List

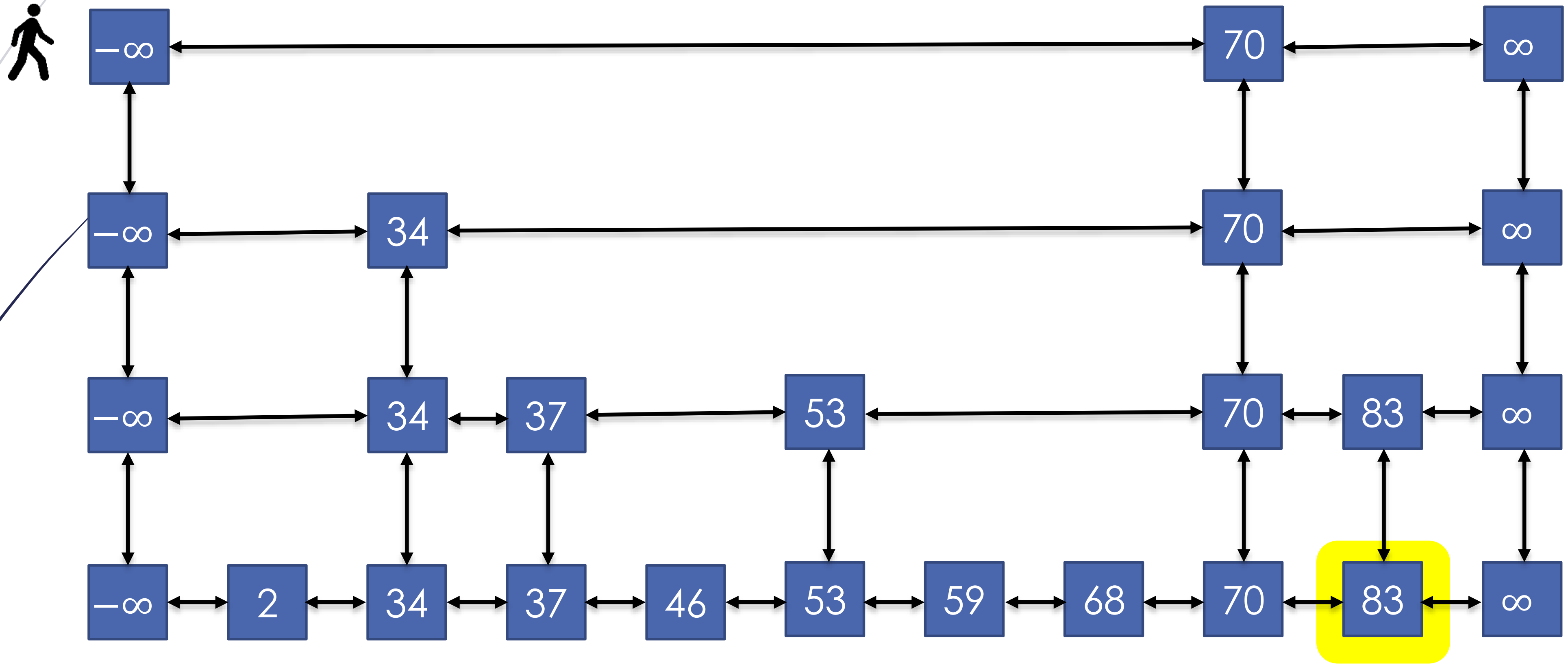
Each node at one of the levels is promoted to the next level with probability ½.



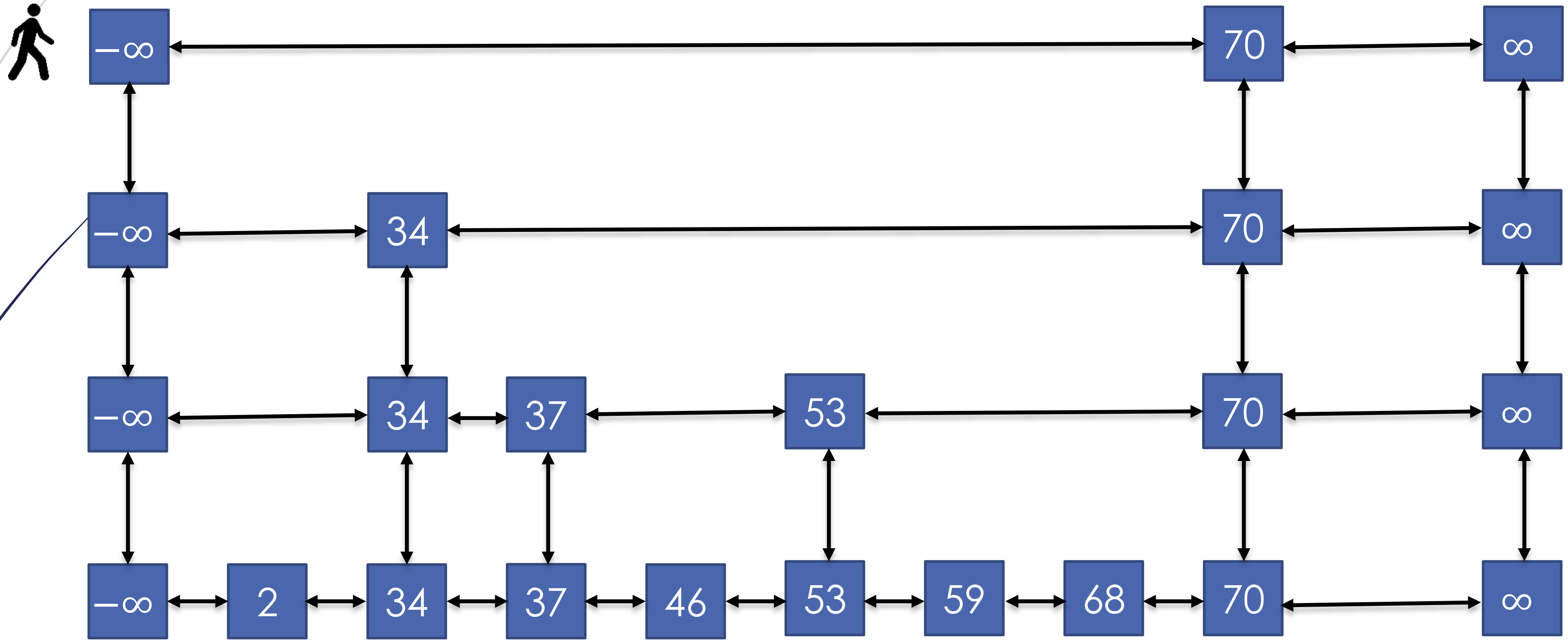
Searching for 68



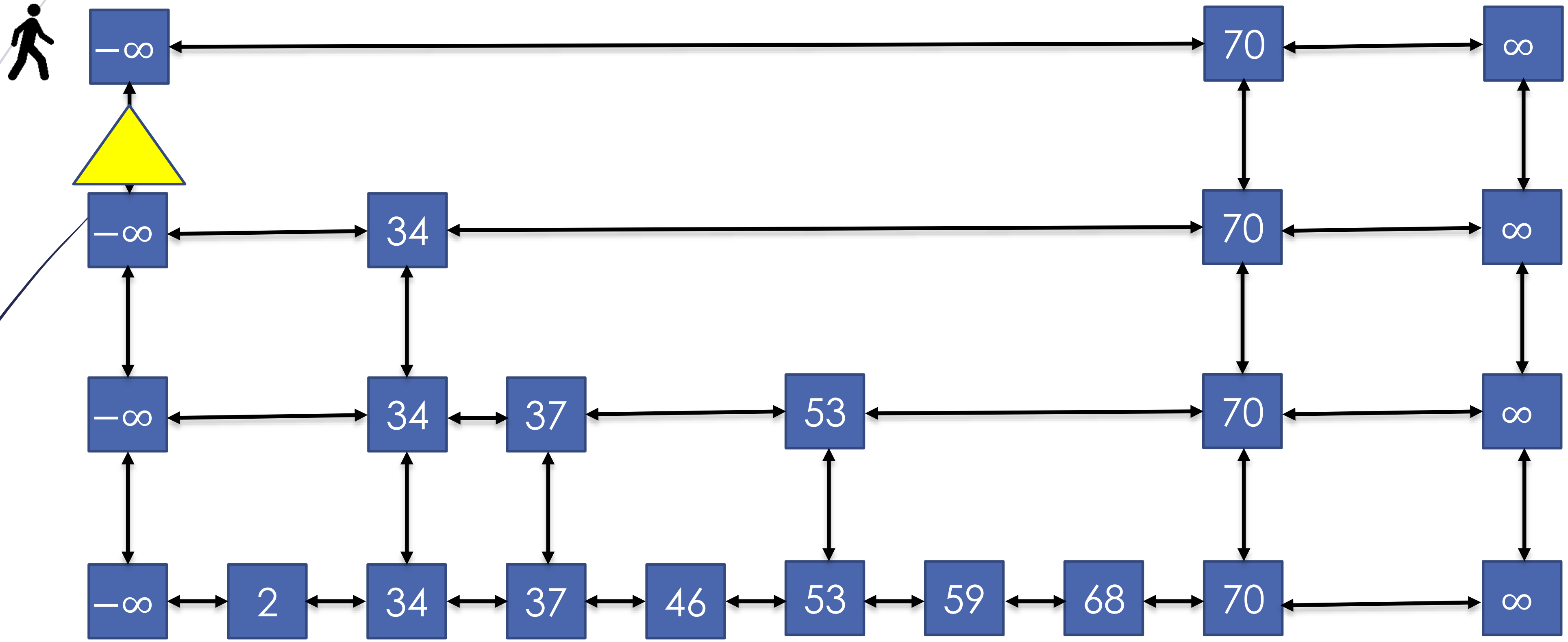
Searching for 83 and Deleting it



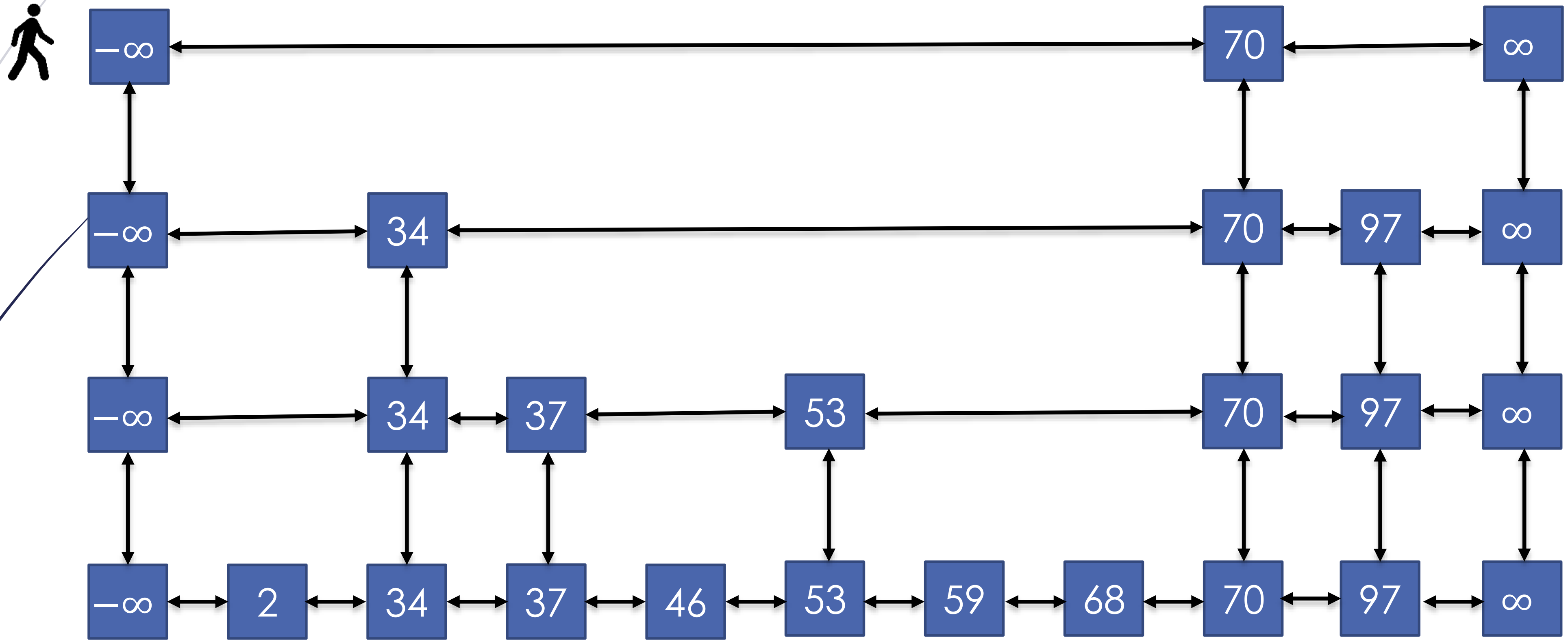
Searching for 83 and Deleting it



Inserting 97



Inserting 97



Analysis of Skip Lists

Number of levels, space complexity, search times

Expected Number of Levels

- Pick an item x. What is the probability that x is raised to level i or more?

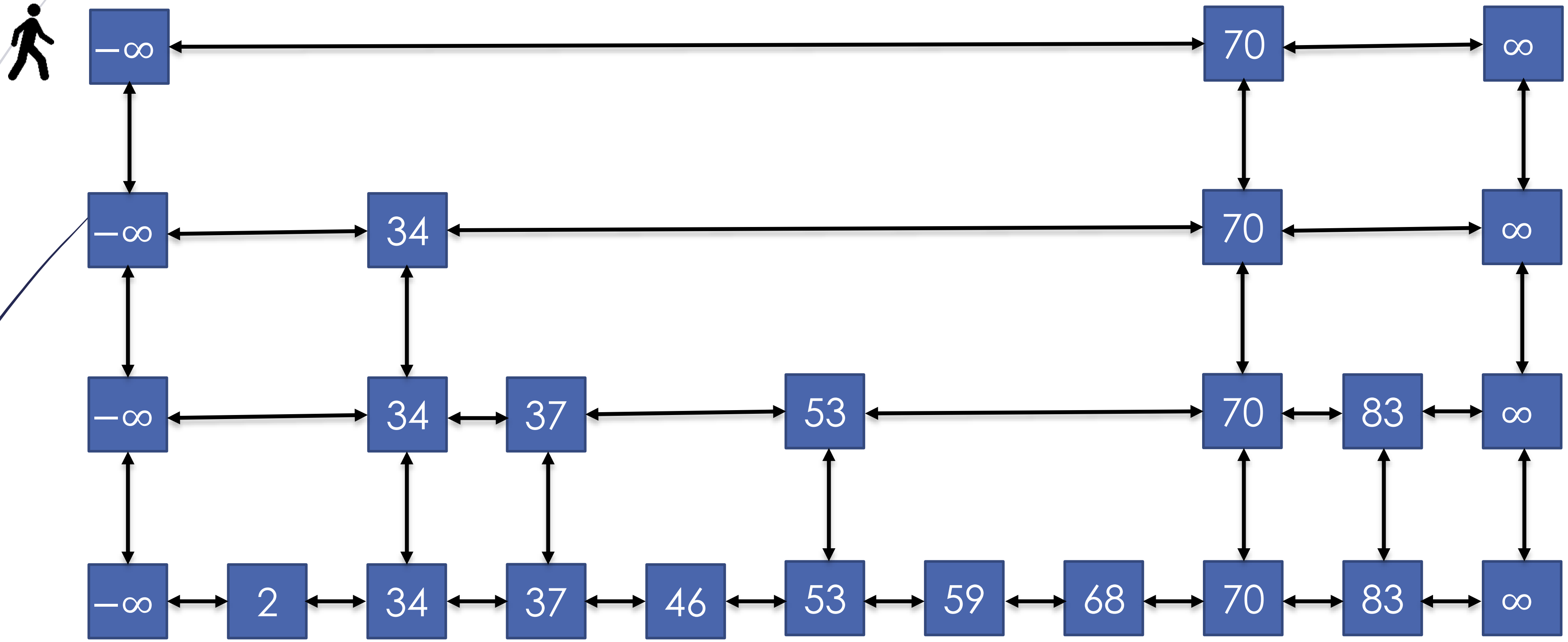
  $\frac{1}{2^i}$ (= prob of fair coin coming heads i times)
- Probability p_i of at least one item at level i or more is $p_i \le \frac{n}{2^i}$ (using the union bound).
- When $i = 3\log n$, $p_i \le \frac{n}{2^{3\log n}} = \frac{n}{n^3} = \frac{1}{n^2}$.
- In other words, the number of levels is less than $3\log n$ WHP.

Expected Space Complexity

- Each item has a constant number of pointers, so we only need to count the number of items.
- Since expected number of items at level i is $\frac{n}{2^i}$, the total number of items is

$$\sum_{i=0}^{\infty} \frac{n}{2^i} = n \sum_{i=0}^{\infty} \frac{1}{2^i} = n\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right) \in O(n).$$

Recall Searching for 68



Search Time

- To search for an item, we alternate between
  1. making forward moves (until we overshoot the searched item) and
  2. stepping down one level.
- There are $O(\log n)$ levels WHP.
- The items encountered at a level do not also occur at the level above (except for the first item and possibly the last item).

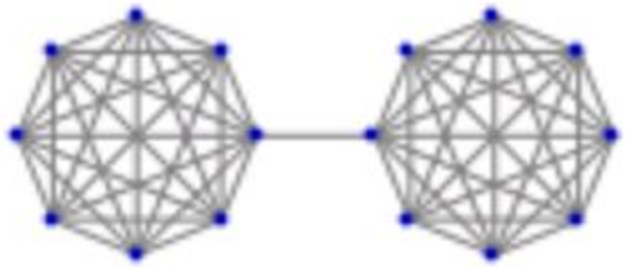
Search Time

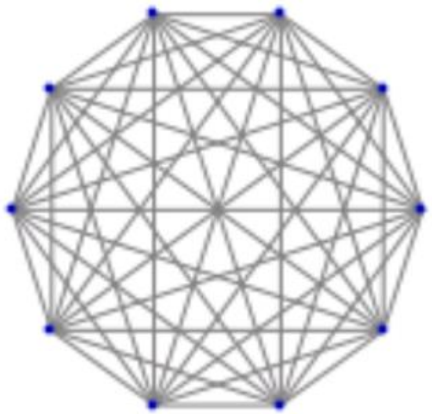
- The probability that an item is not raised one more level is ½.
- Thus, expected number of items encountered at a level is at most $2 + 2 = 4$.
- Thus expected search time is $O(\log n)$.
- Exercise: Show that insertion and deletion also take expected $O(\log n)$ time.

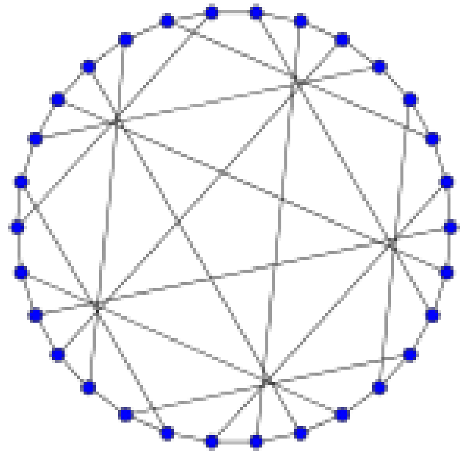
Karger's Mincut Algorithm

The Mincut problem

Which of these graphs will be a robust network?







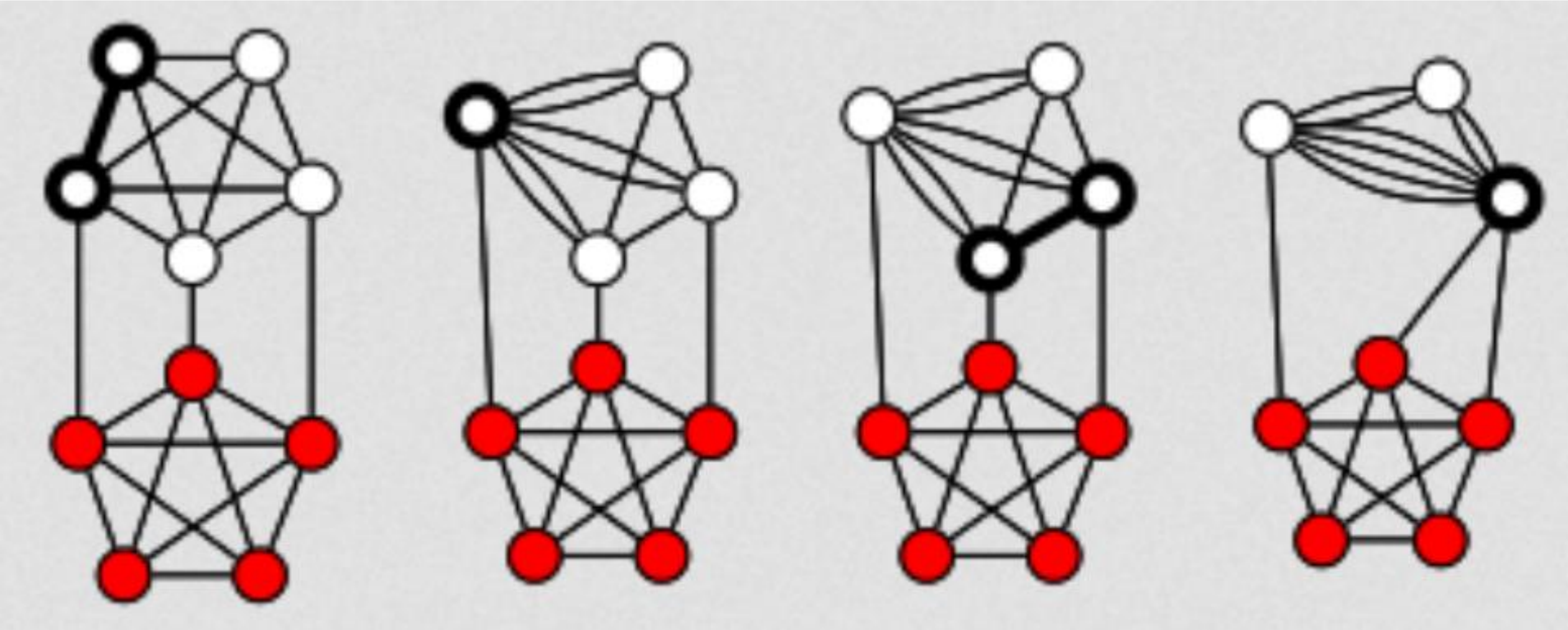
The Mincut problem

- Consider an undirected, unweighted graph $G = (V, E)$.
- A cut is a partition of V into two sets S and $V \setminus S$.
- The cutset corresponding to this cut is the set of edges with one end in S and the other in $V \setminus S$.
- Our goal is to find the partition such that the corresponding cutset has least cardinality.

Karger's Mincut Algorithm

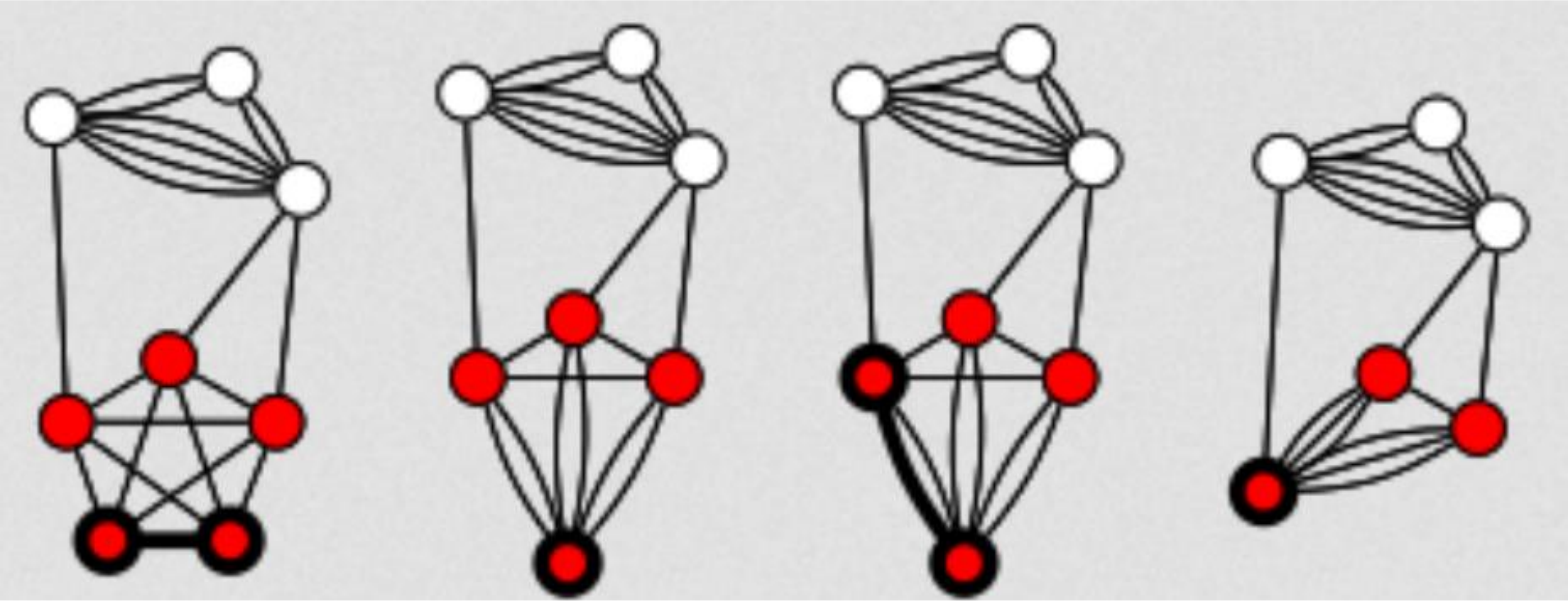
- There is a flow based algorithm, but the randomized algorithm proposed by Karger is very elegant.
- Repeat until graph has only two vertices
  - Pick an edge (u, v) uniformly at random from all edges.
  - Contract (u, v), i.e., coalesce u and v into one. Remove self loops, but retain multi-edges
- Let S be the set of original vertices that lead to one of the final two vertices.
- **Lemma**: with probability at least $\frac{1}{\binom{n}{2}}$, S and $V \setminus S$ induce the smallest cutset.

Karger’s Mincut Algorithm Illustration



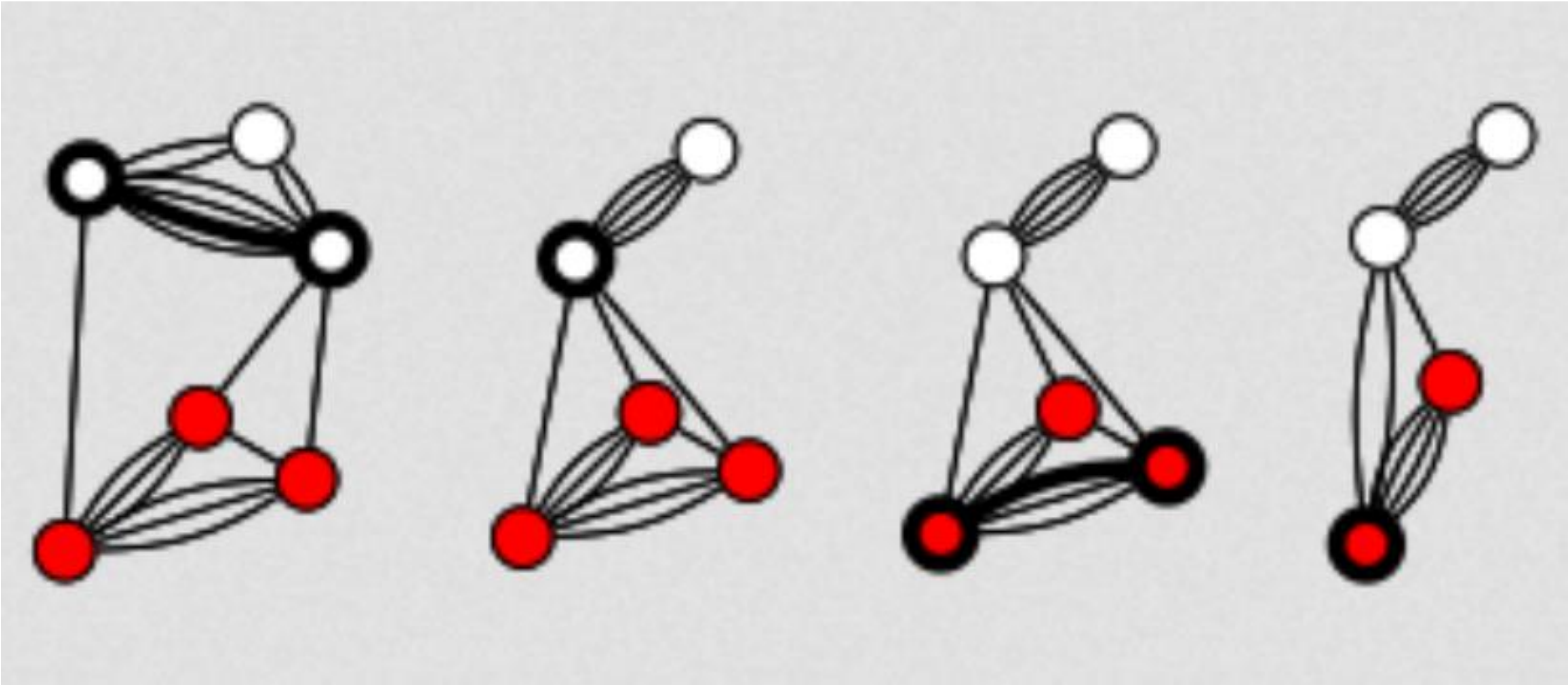
Karger’s Mincut Algorithm. Source: https://en.wikipedia.org/wiki/Karger's_algorithm

Karger’s Mincut Algorithm Illustration



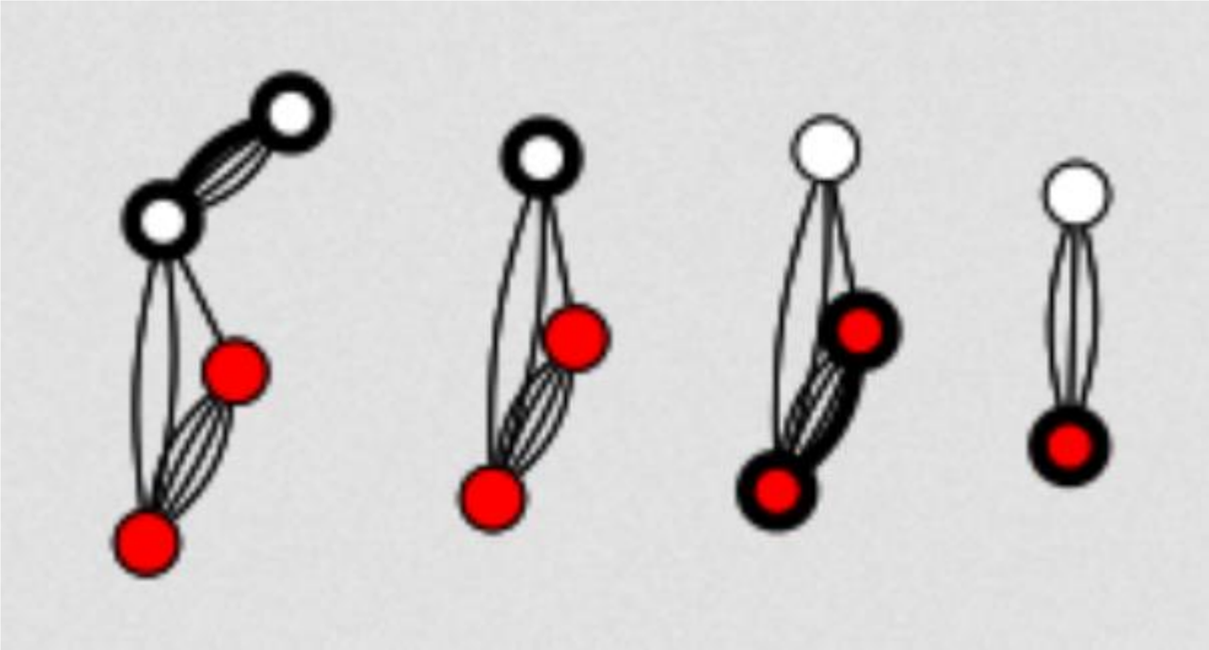
Karger’s Mincut Algorithm. Source: https://en.wikipedia.org/wiki/Karger's_algorithm

Karger’s Mincut Algorithm Illustration



Karger’s Mincut Algorithm. Source: https://en.wikipedia.org/wiki/Karger's_algorithm

Karger’s Mincut Algorithm Illustration



Karger’s Mincut Algorithm. Source: https://en.wikipedia.org/wiki/Karger's_algorithm

Analysis

- Let C^* be a cutset of minimum cardinality.
- In order to prove the lemma, we will simply prove that:

$$\Pr[\text{cut produced by algorithm is } C^*] \geq \frac{1}{\binom{n}{2}}.$$

- Let $e_1, e_2, \ldots, e_{n-2}$ be the sequence of edges contracted by the algorithm. The algorithm succeeds if none of them are in C^*.

Minimum Degree and Cutset Size

- Let $k = |C^*|$. Clearly, the minimum degree of G must be at least k.
- Moreover, this holds as an invariant for all intermediate (multi) graphs.
  - The edges incident at the low degree vertex of the intermediate graph correspond to a cutset in the original graph.

Analysis (contd.)

- Let G_j denote the graph after j contractions.
- Note: G_j has $n - j$ vertices
- The number of edges, therefore, is at least $\frac{(n-j)k}{2}$.

$$\Pr[\text{cut produced by algorithm is } C^*]$$
$$= \Pr[e_1 \notin C^*] \prod_{j=1}^{n-3} \Pr[e_{j+1} \notin C^* | e_1 \dots e_j \notin C^*]$$

Analysis (contd.)

$$\geq \prod_{j=0}^{n-3}\left(1 - \frac{k}{\frac{(n-j)k}{2}}\right)$$

$$= \frac{n-2}{n} \times \frac{n-3}{n-1} \times \cdots \times \frac{2}{4} \times \frac{1}{3}$$

$$= \frac{2}{n(n-1)} = \frac{1}{\binom{n}{2}}. \qquad \blacksquare$$

Boosting the Probability of Success

- To improve our success probability, we can simply repeat the algorithm, say, $c \log n$ times (for any $c > 0$) and report the smallest cutset.
- Probability that all these repetitions will fail is at most

$$\left(1 - \frac{1}{\binom{n}{2}}\right)^{c\,\binom{n}{2}\log_e n} \leq \frac{1}{n^c}.$$

- Thus, algorithm will succeed WHP.

Running Time Analysis

- This is a Monte Carlo randomized algorithm, i.e., the running time is deterministic, but the algorithm may fail to produce the correct mincut with very small probability.
- Exercise: show that each run of Karger's algorithm takes $O(n^2)$ time.
- Thus, with probability boosting, the algorithm takes $O(n^4 \log n)$ time.

Thank You!