External-Memory Bidirectional Search

Nathan R. Sturtevant and Jingwei Chen
University of Denver

Grant 1551406

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Background/Motivation

- Holte et. al., 2016 refined theory about bidirectional search from Barker & Korf, 2015
  - Introduced a new bidirectional search algorithm
  - Needed/wanted to solve even larger problems
  - Insufficient RAM

External Memory

- External Memory
  - Hard disk or SSD
- Properties

External Memory

- External Memory
  - Hard disk or SSD

- Properties
  - High latency for random access

External Memory

- External Memory
  - Hard disk or SSD

- Properties
  - High latency for random access
  - High throughput

External Memory

- External Memory
  - Hard disk or SSD

- Properties
  - High latency for random access
  - High throughput
  - High storage (compared to main memory)

External Memory Search

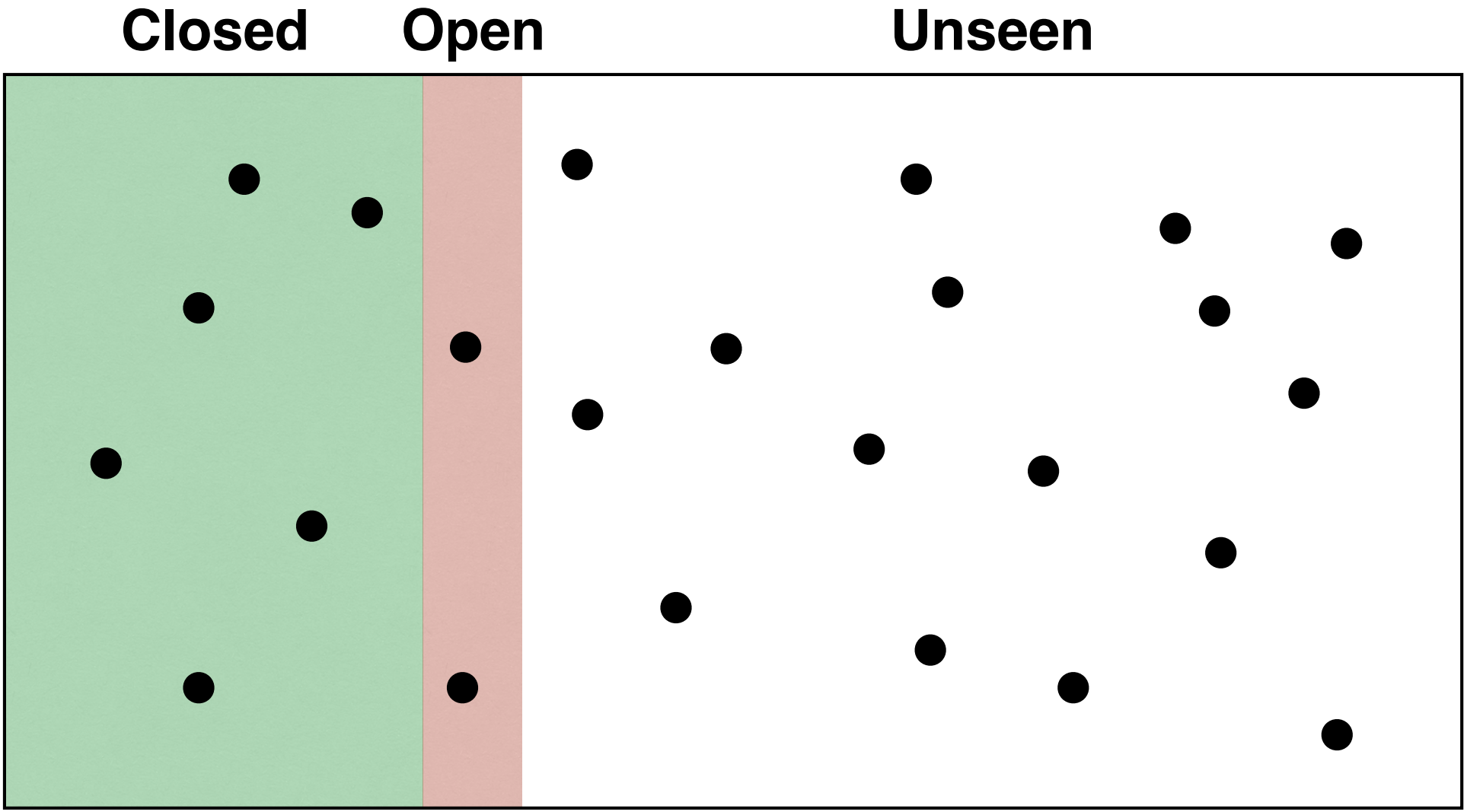
- External memory searches have been well-studied for unidirectional search
  - Delayed Duplicate Detection (Korf et al)
    - Hash-based
    - Sorting based
  - Structured Duplicate Detection (Zhou et al)

Contributions

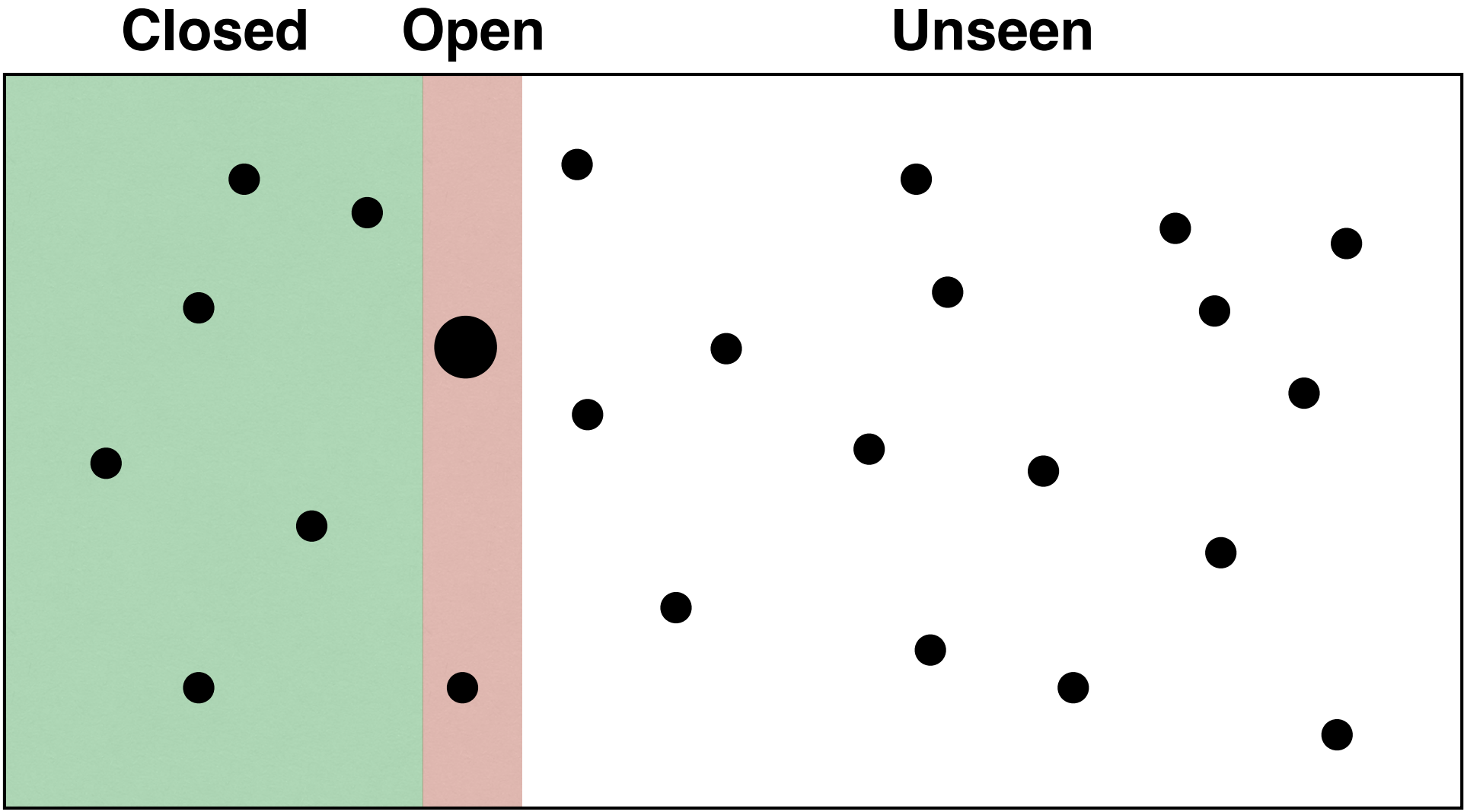
- Bidirectional external memory search
  - Delayed Solution Detection
- Experimental results

Best First Search

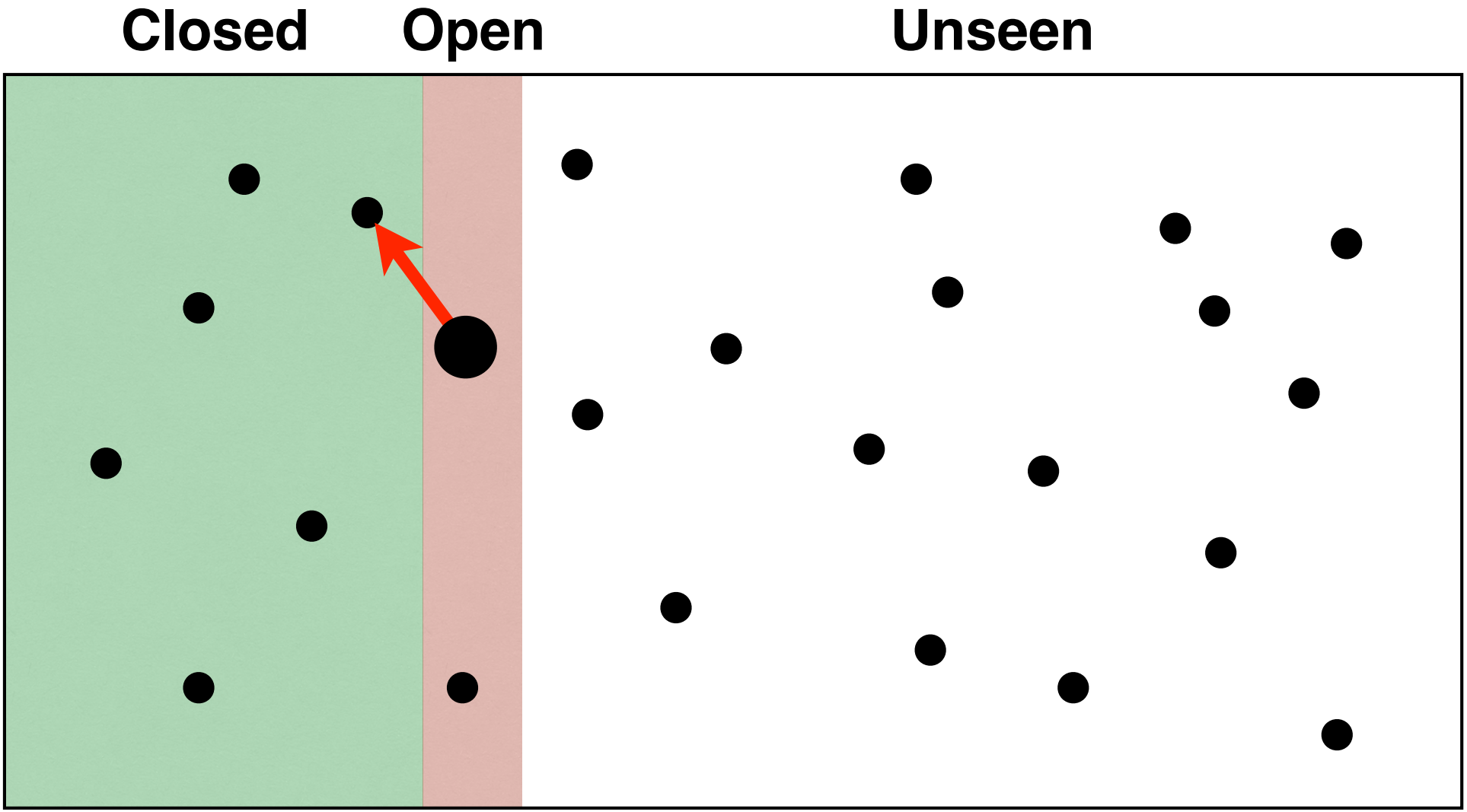
Best-First Search



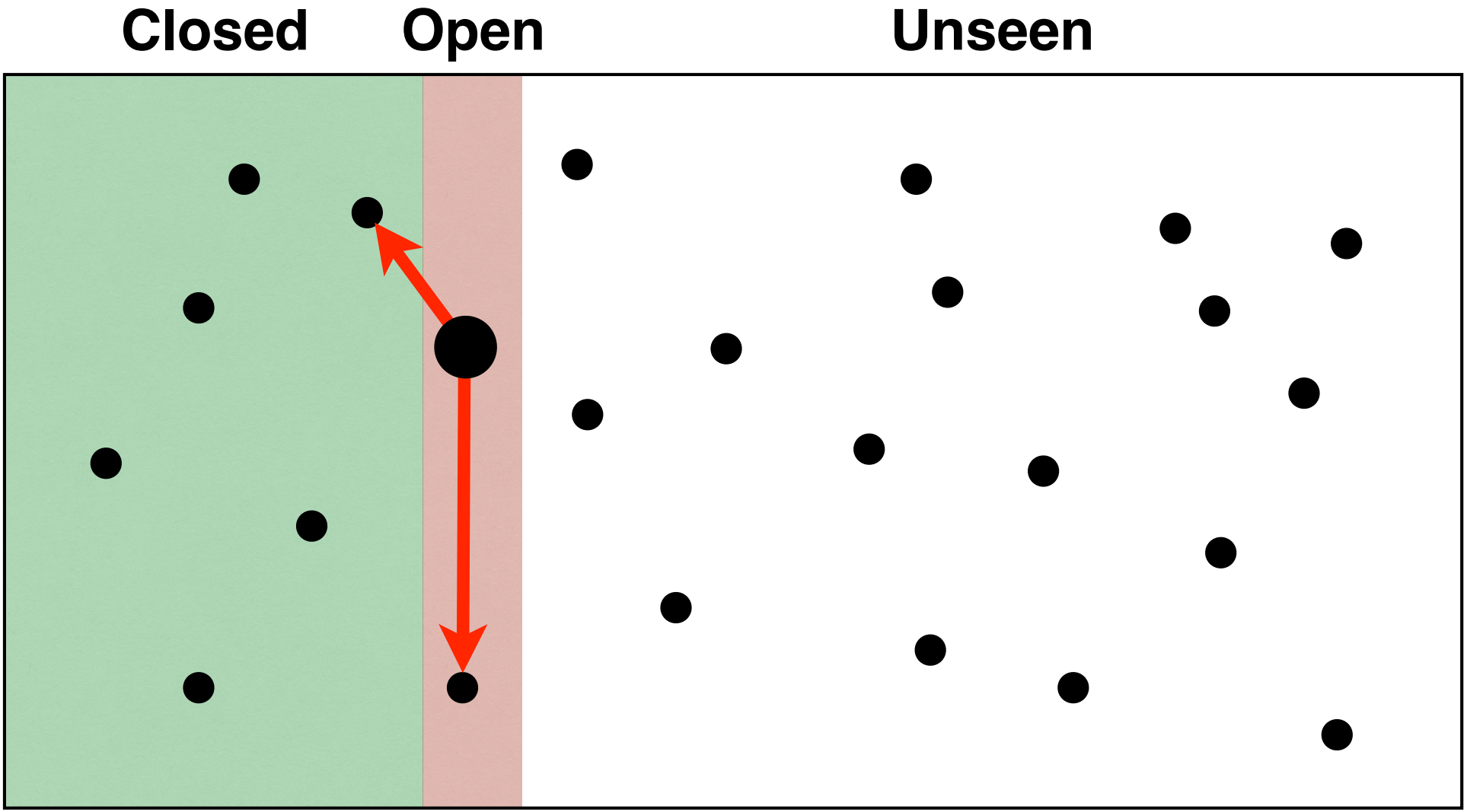
Best-First Search



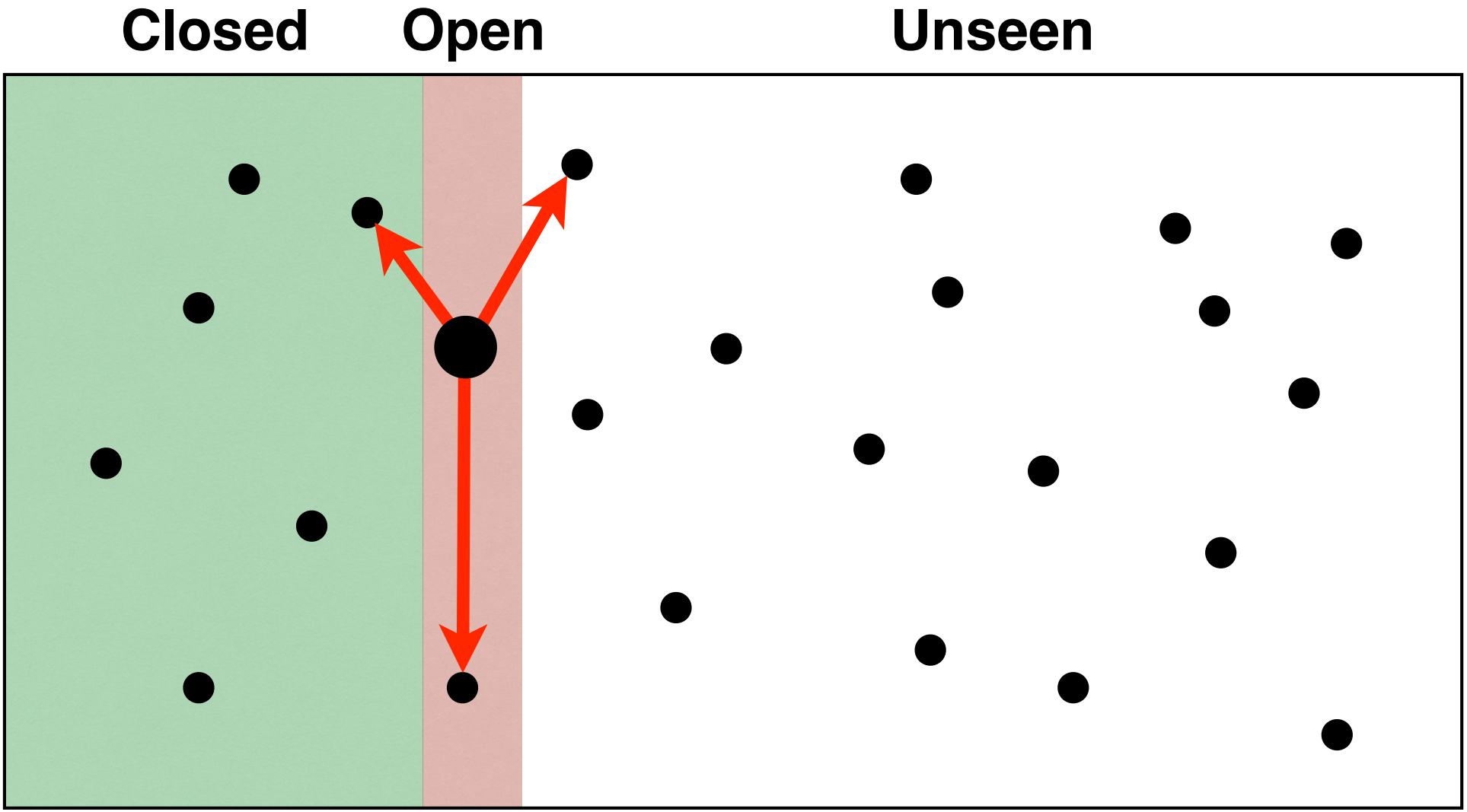
Best-First Search



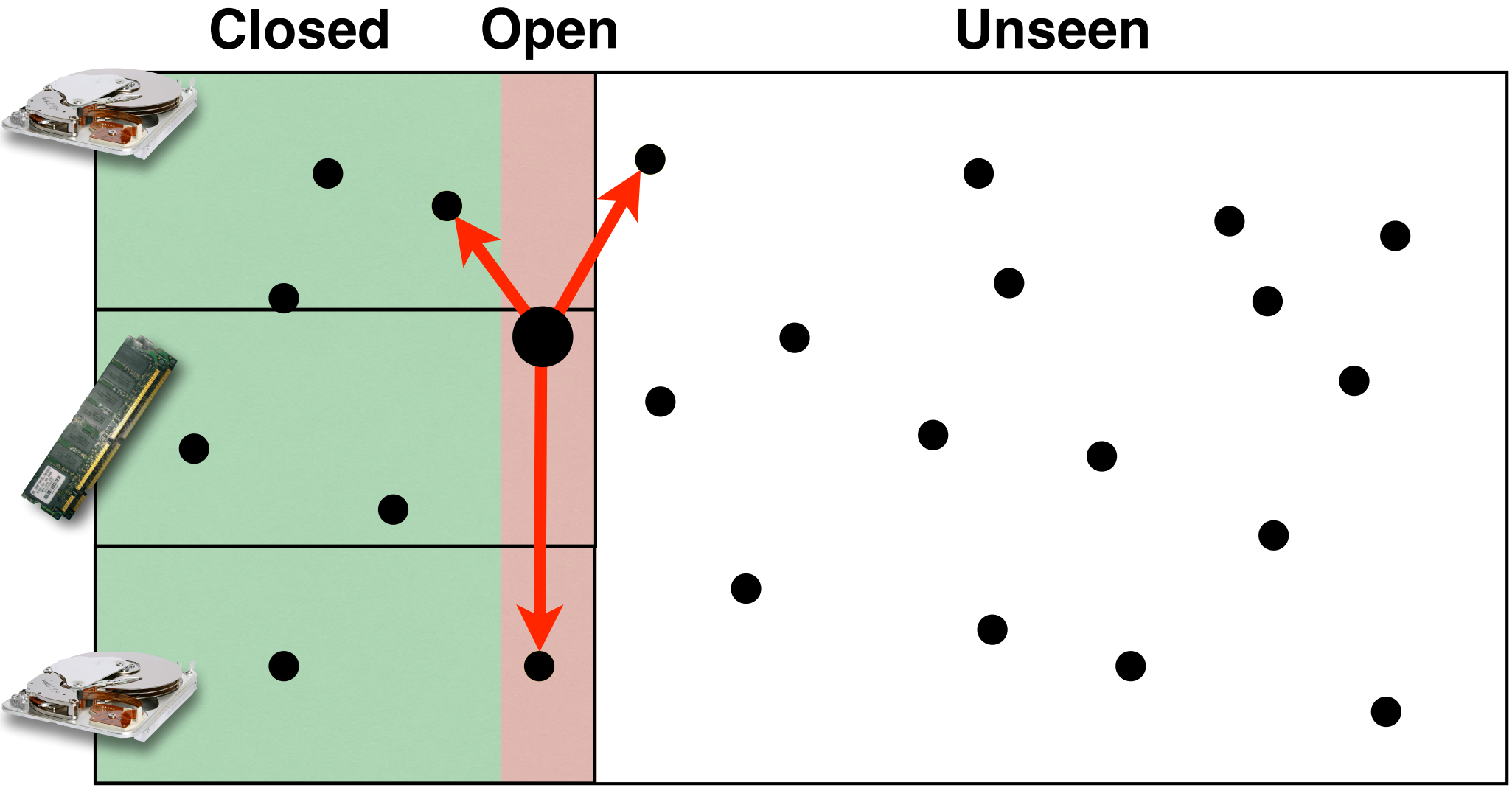
Best-First Search



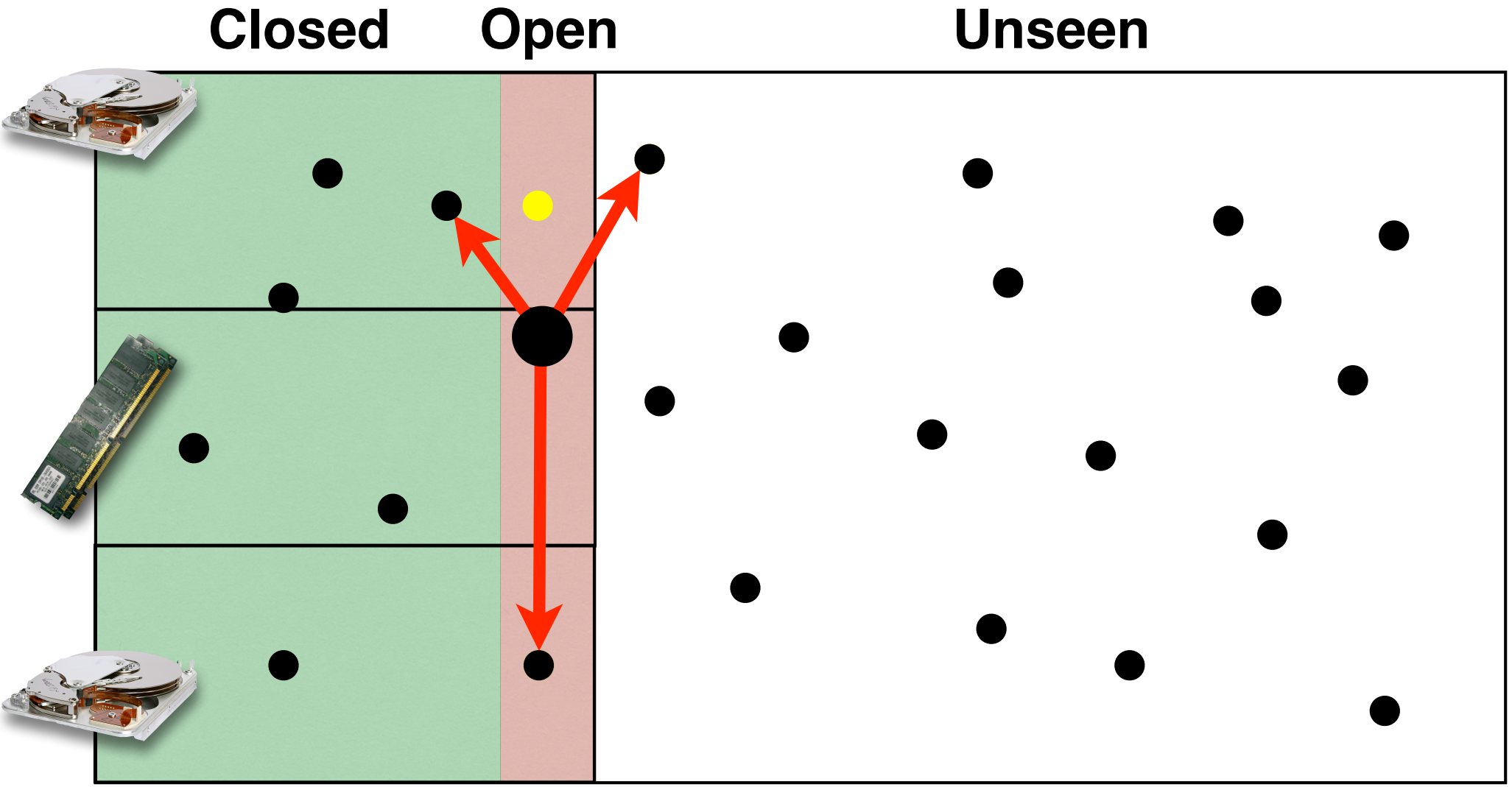
Best-First Search



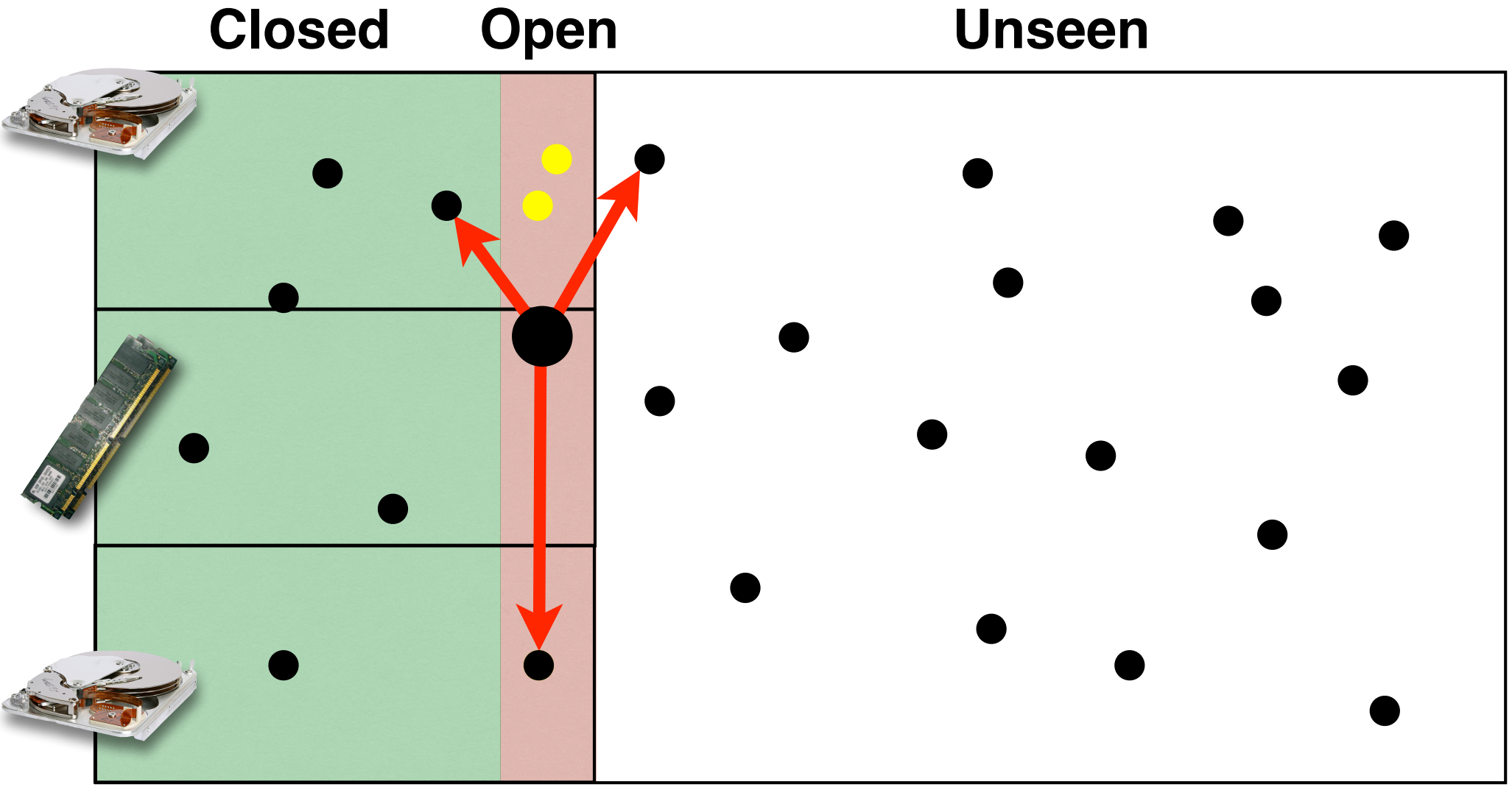
External-Memory Best-First Search



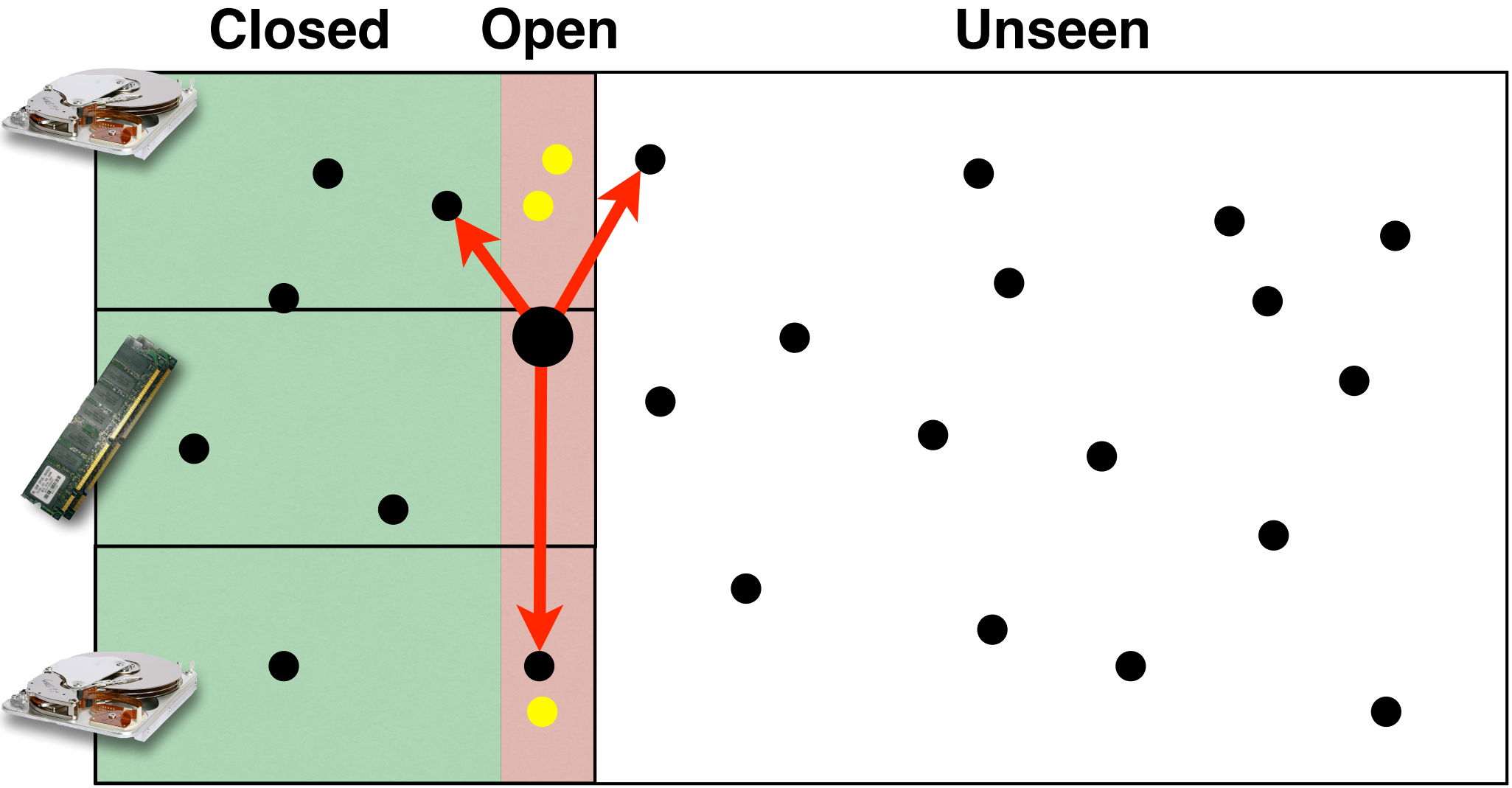
External-Memory Best-First Search



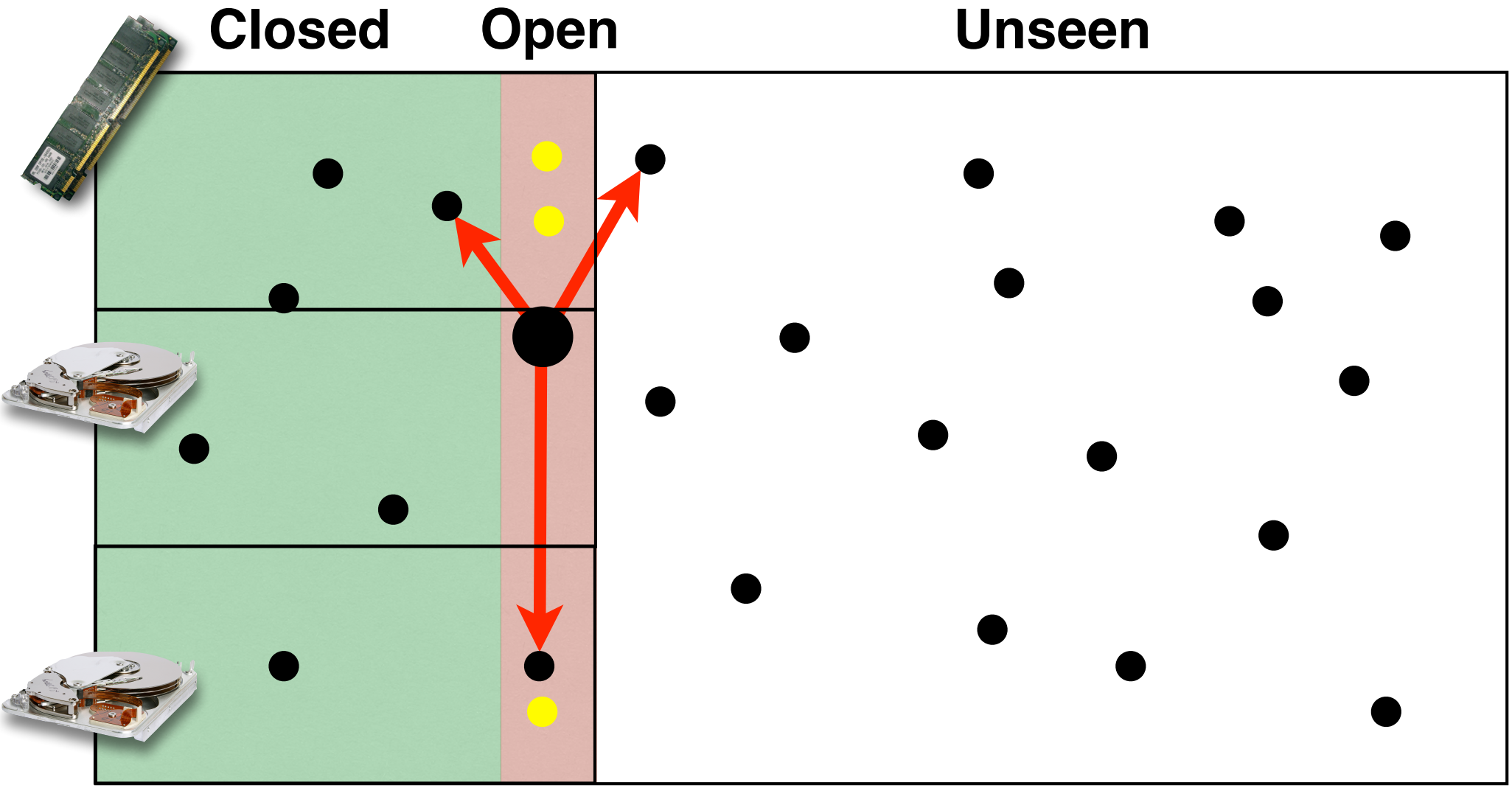
External-Memory Best-First Search



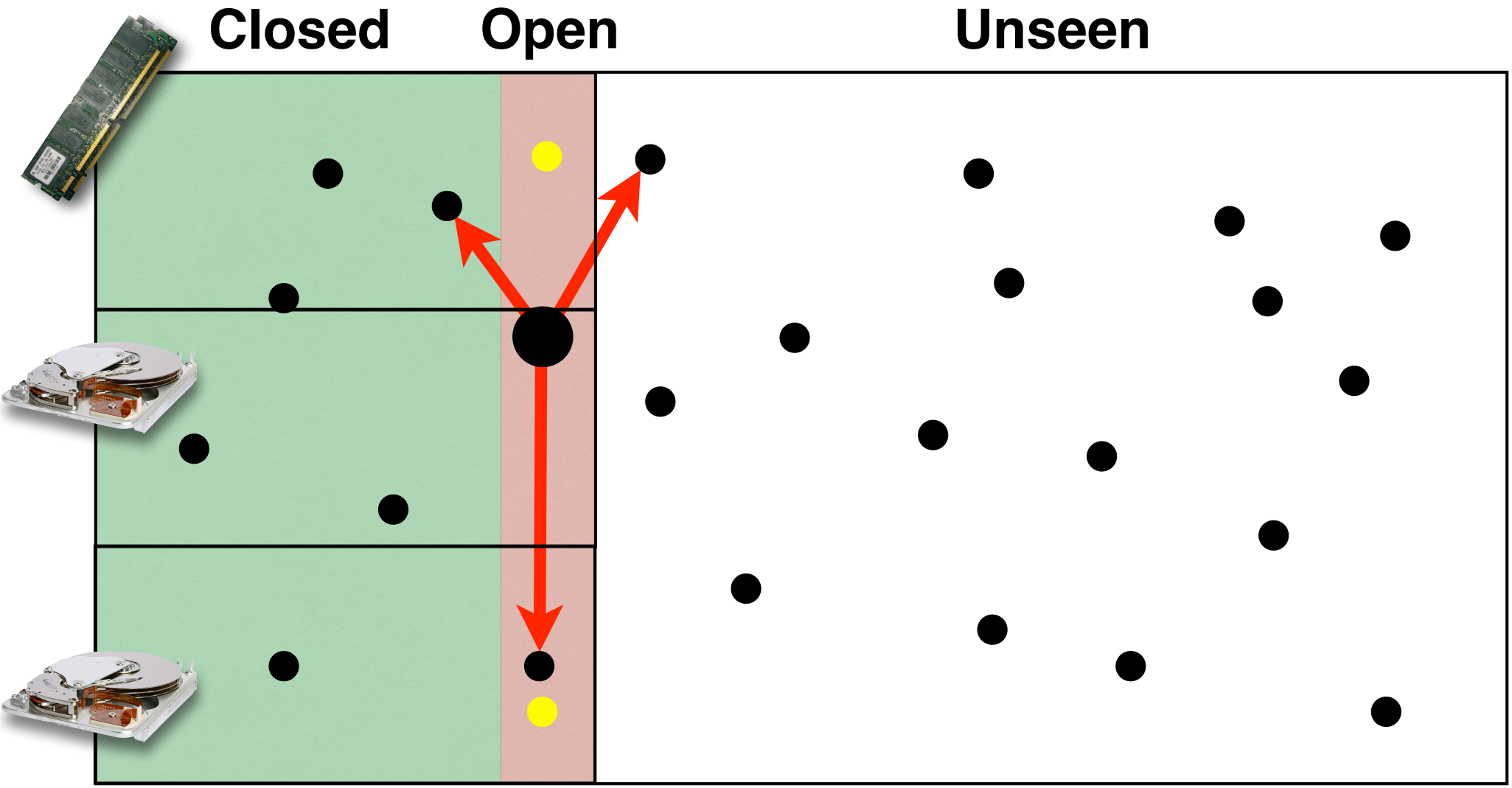
External-Memory Best-First Search



External-Memory Best-First Search



External-Memory Best-First Search



Search Operations

- Remove from OPEN
  - Check if state is a goal
  - Generate successors
    - Check for duplicates in OPEN
    - Check for duplicates in CLOSED
    - Add to OPEN
  - Add state to CLOSED

Search Operations

- Remove from OPEN
  - Check if state is a goal
  - Generate successors
    - Check for duplicates in OPEN
    - Check for duplicates in CLOSED
    - Add to OPEN
  - Add state to CLOSED

Buckets

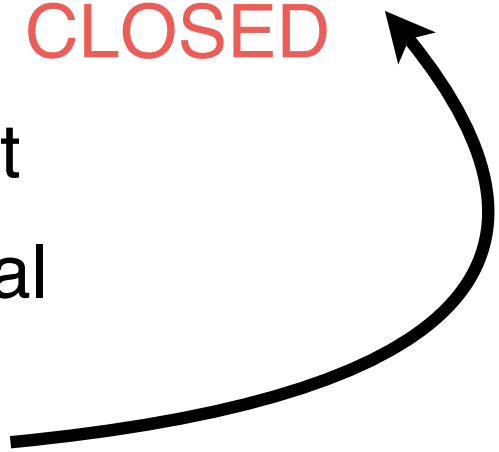
- Divide state space up into buckets:
  - Each bucket should fit into RAM
- Divide based on features such as:
  - f-cost, g-cost, h-cost, hash function

External Memory Search

- Load best bucket into memory (removed dups)
  - Check for duplicates on CLOSED
  - For each state in bucket
    - Check if state is a goal
    - Generate successors
      - Add successors to OPEN

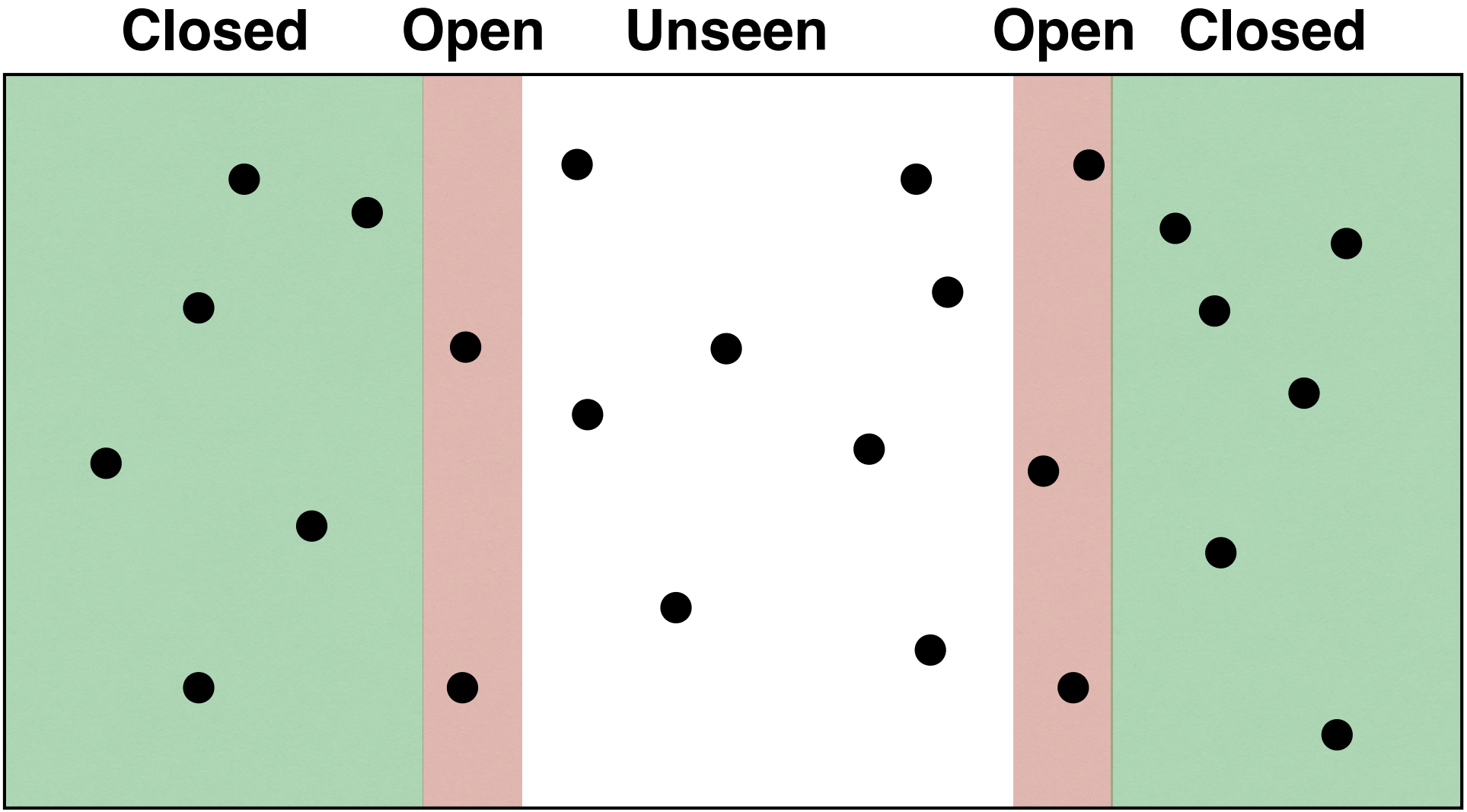
External Memory Search

- Load best bucket into memory (removed dups)
  - Check for duplicates on CLOSED
  - For each state in bucket
    - Check if state is a goal
    - Generate successors
      - Add successors to OPEN

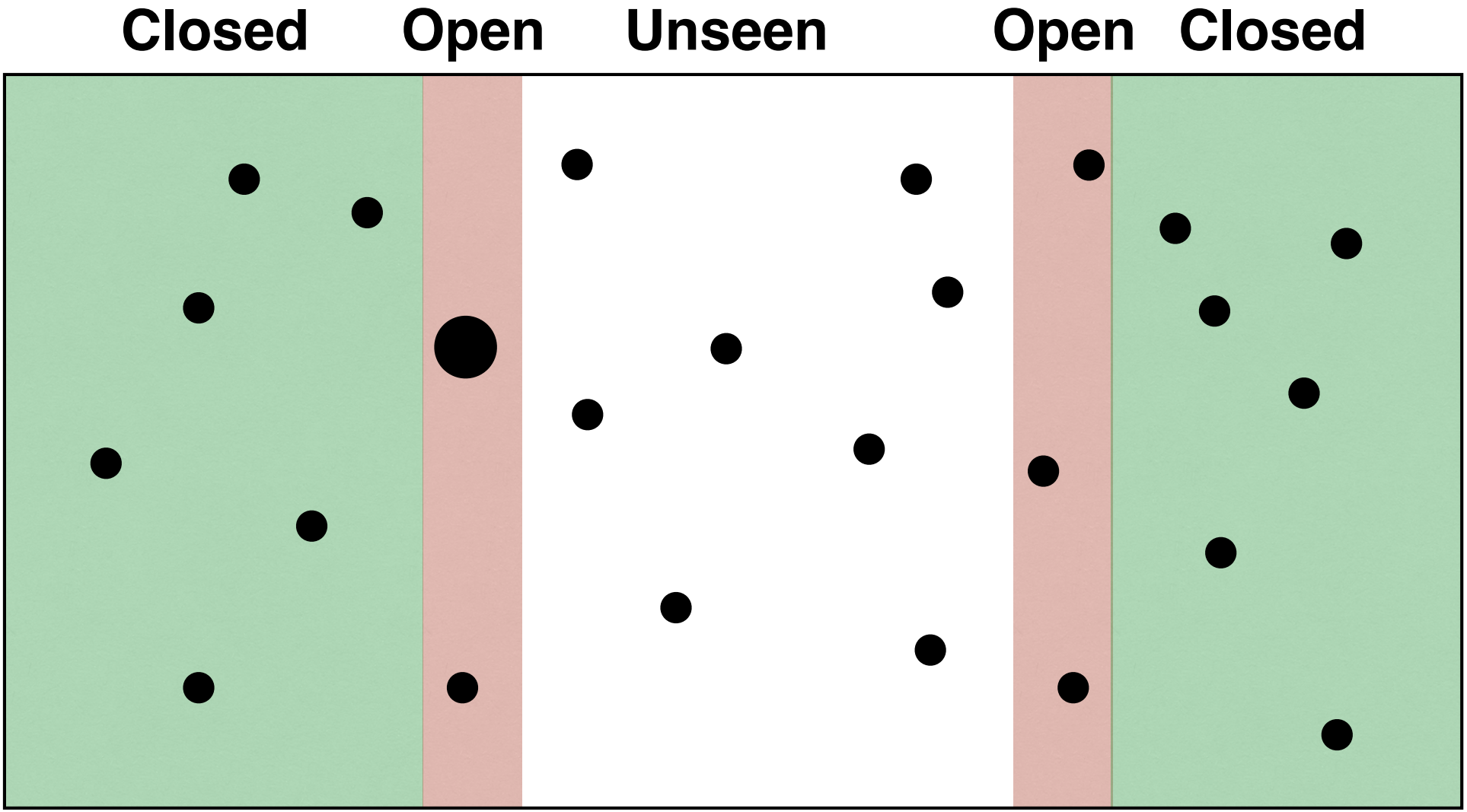


Bidirectional Search

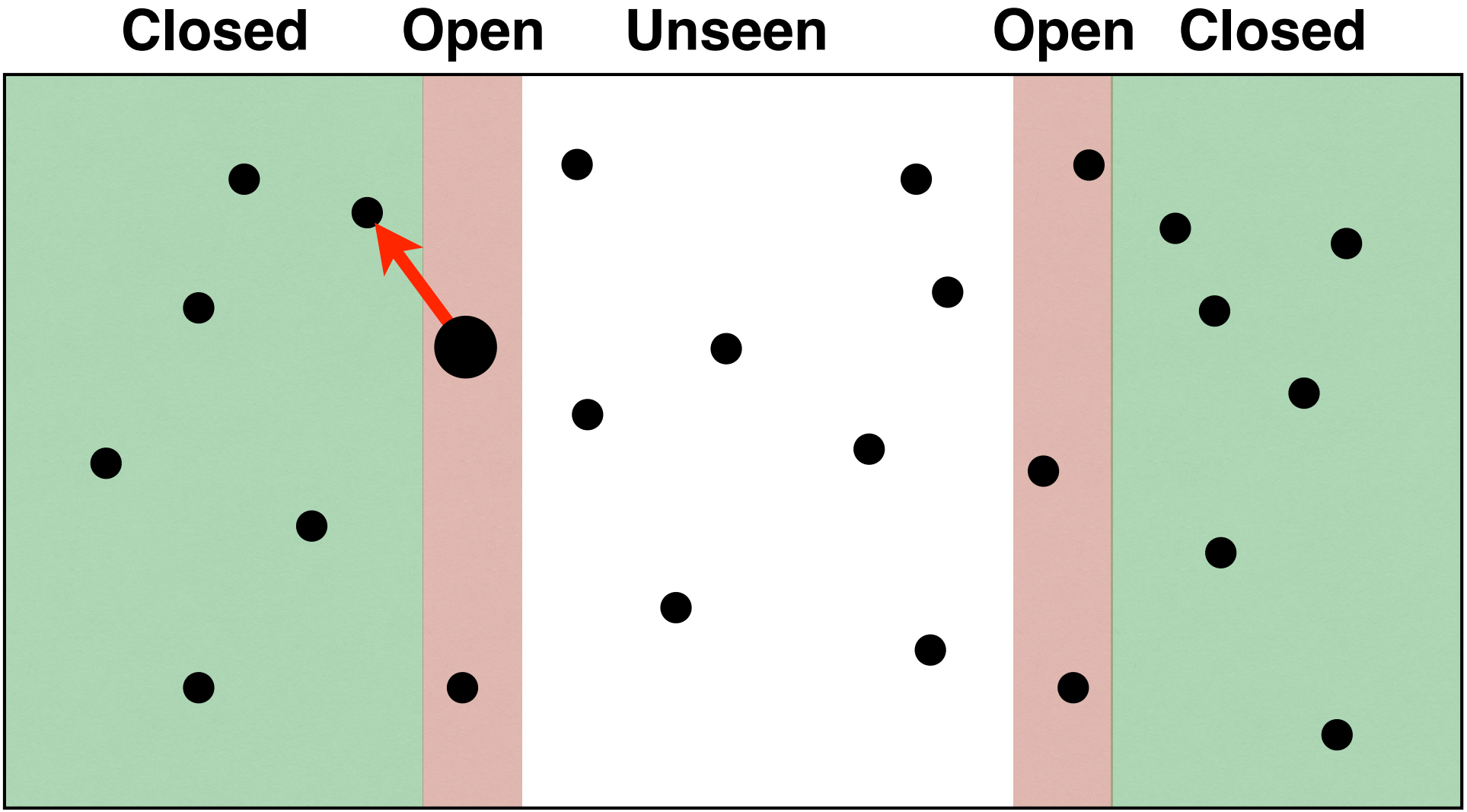
Bidirectional Search



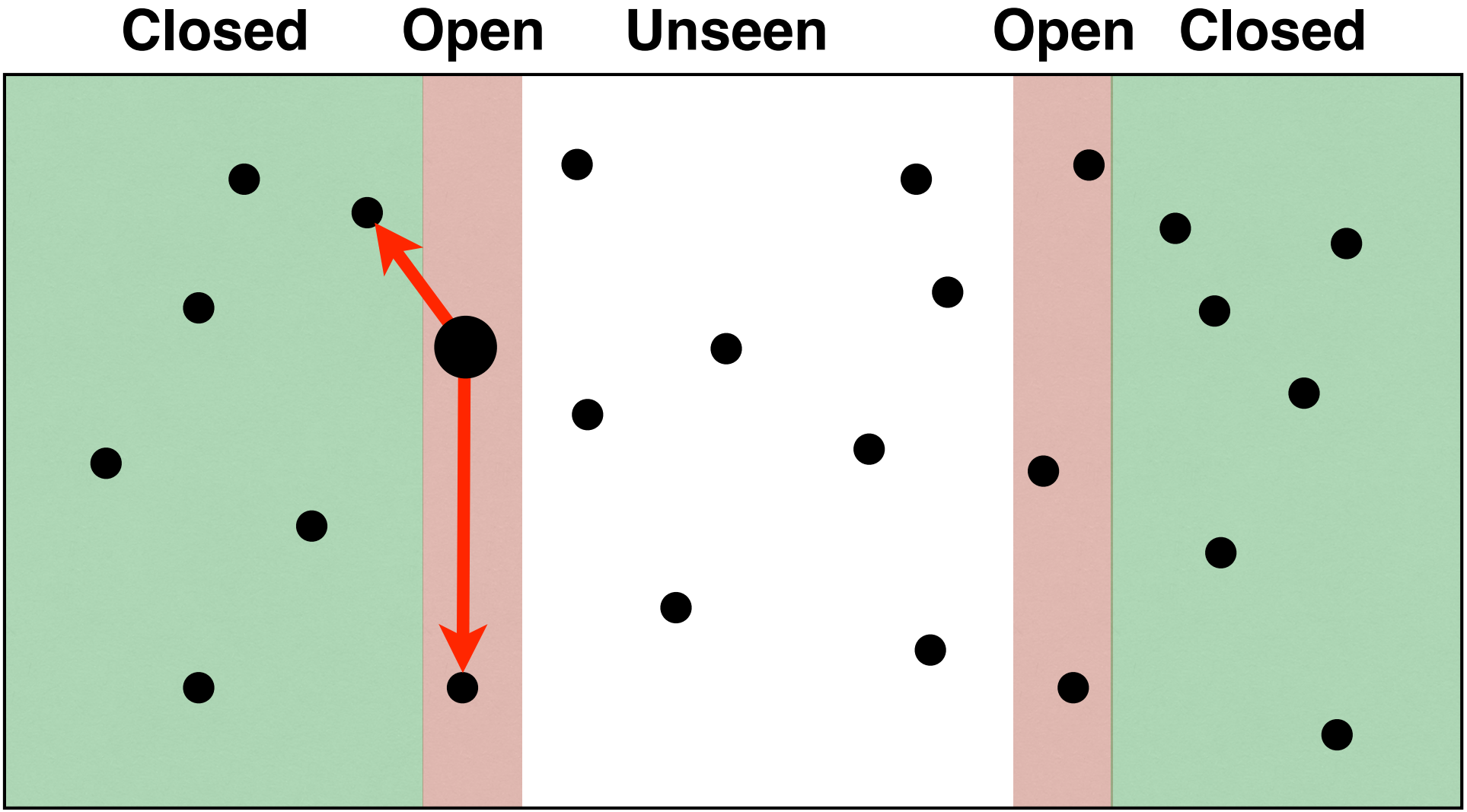
Bidirectional Search



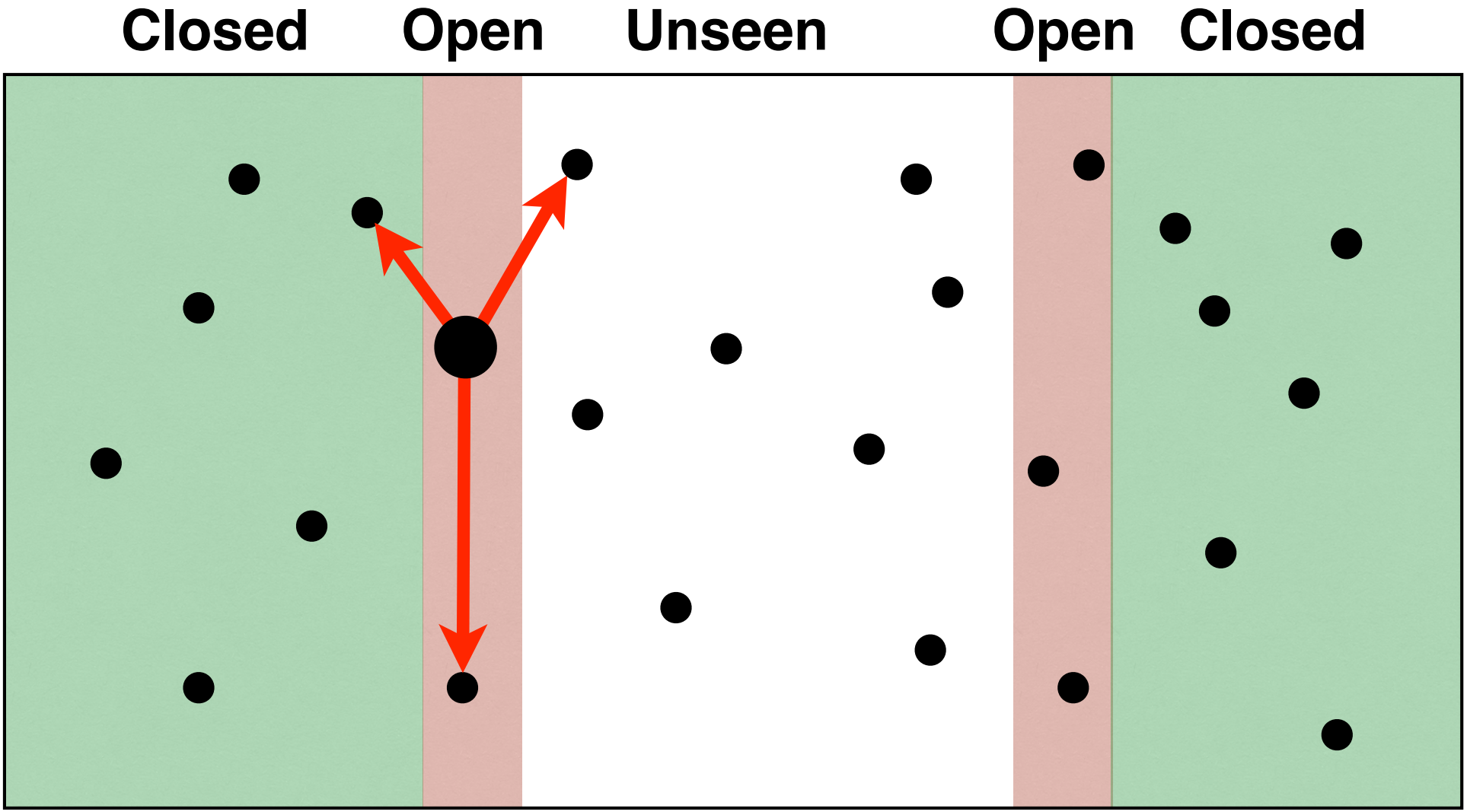
Bidirectional Search



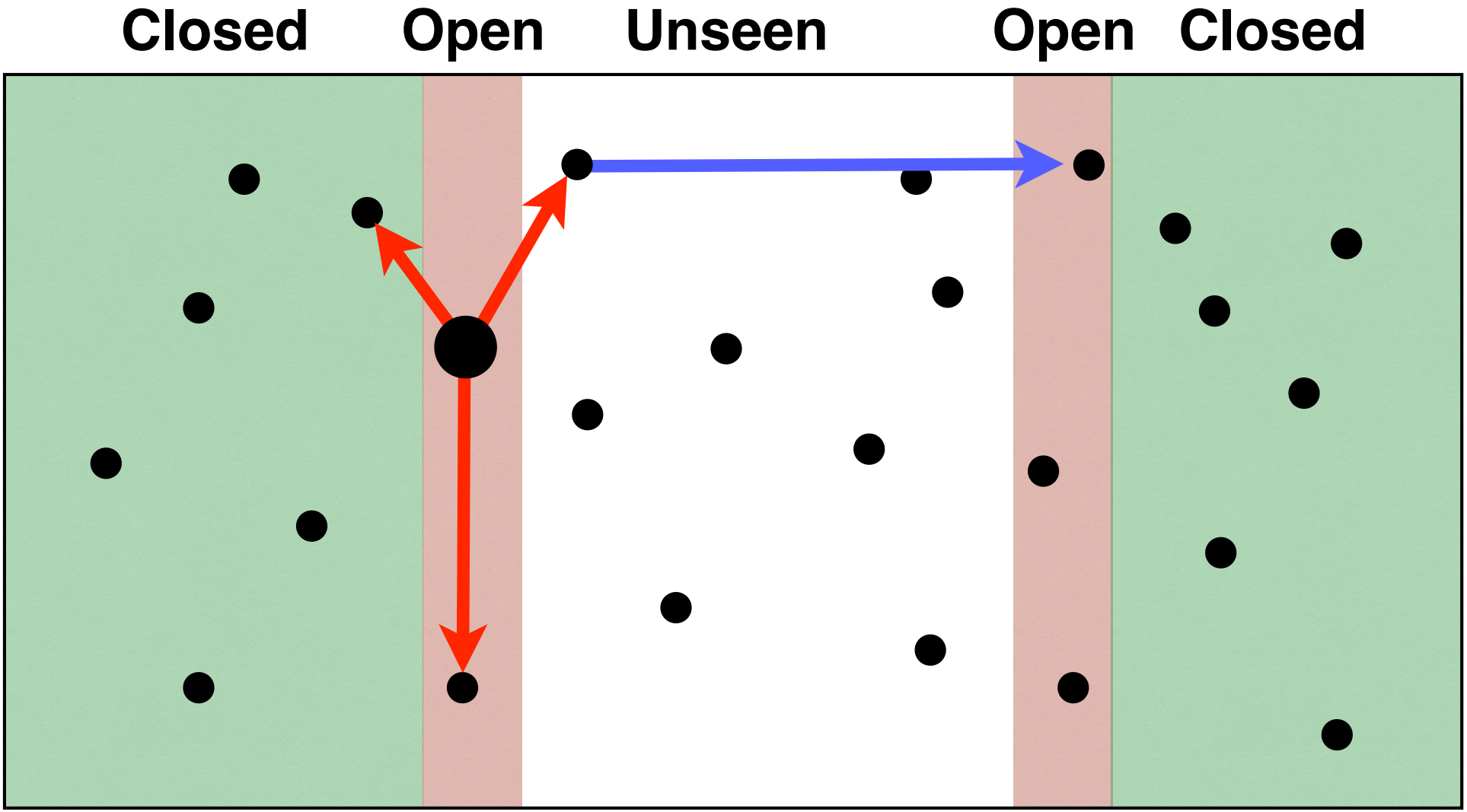
Bidirectional Search



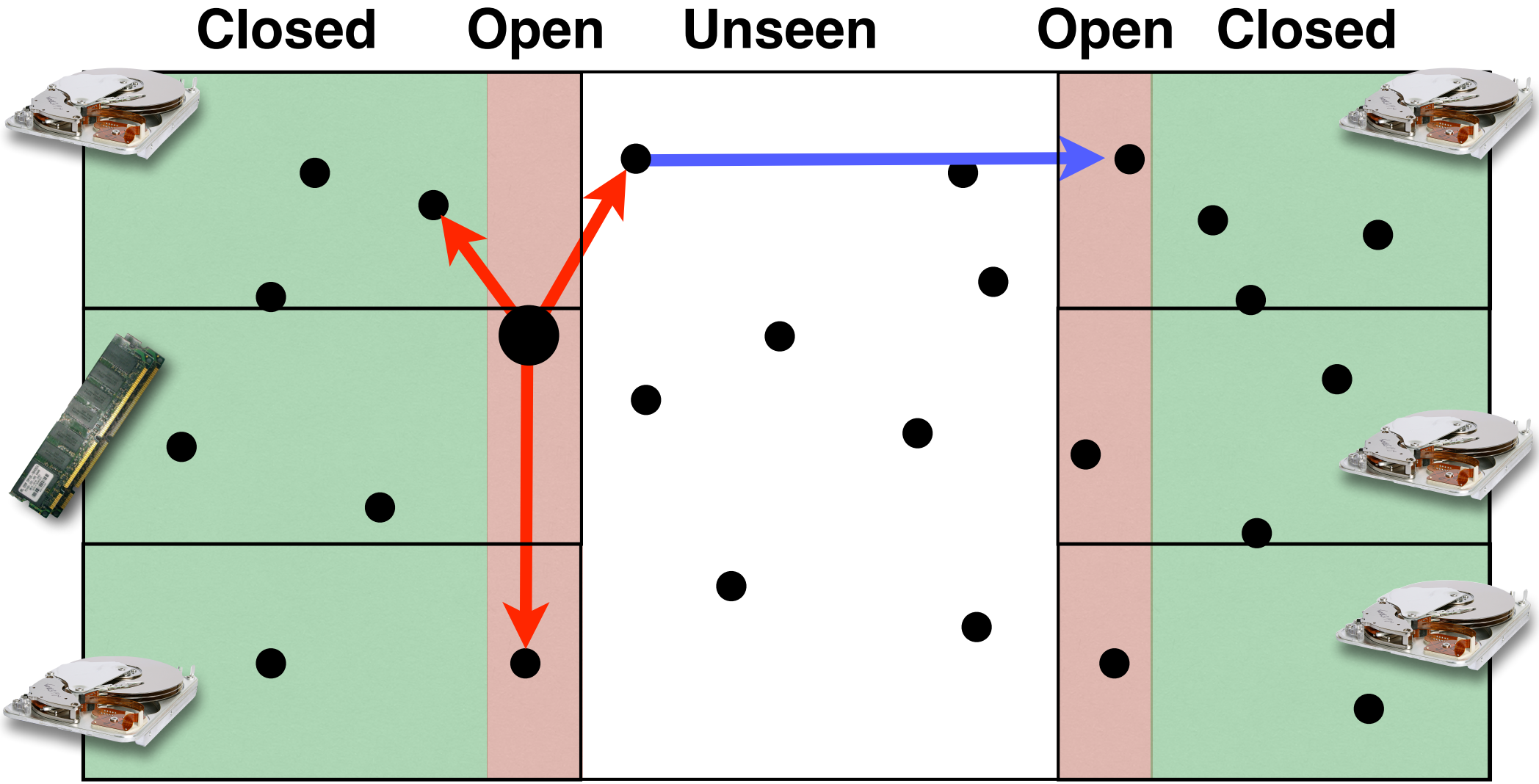
Bidirectional Search



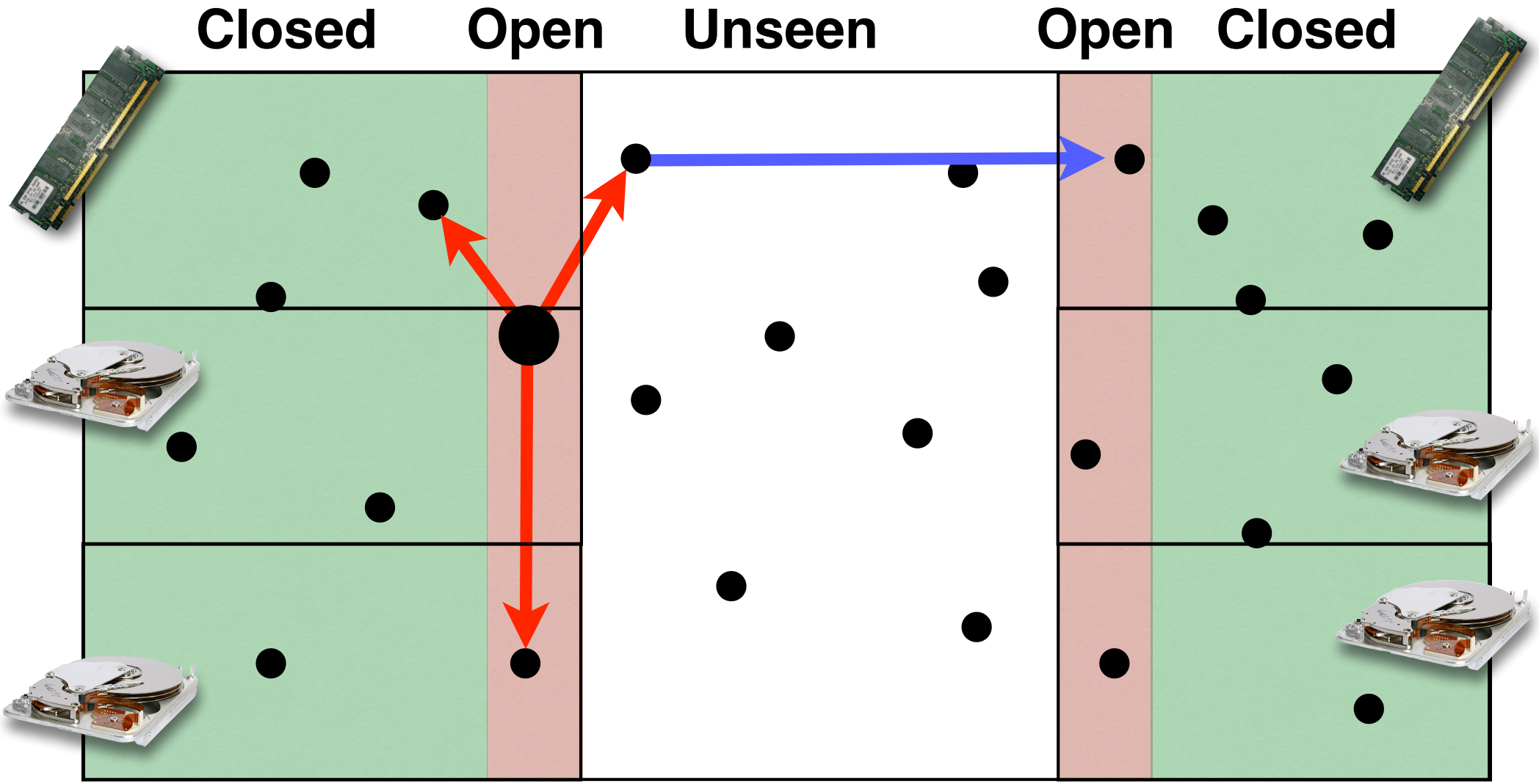
Bidirectional Search



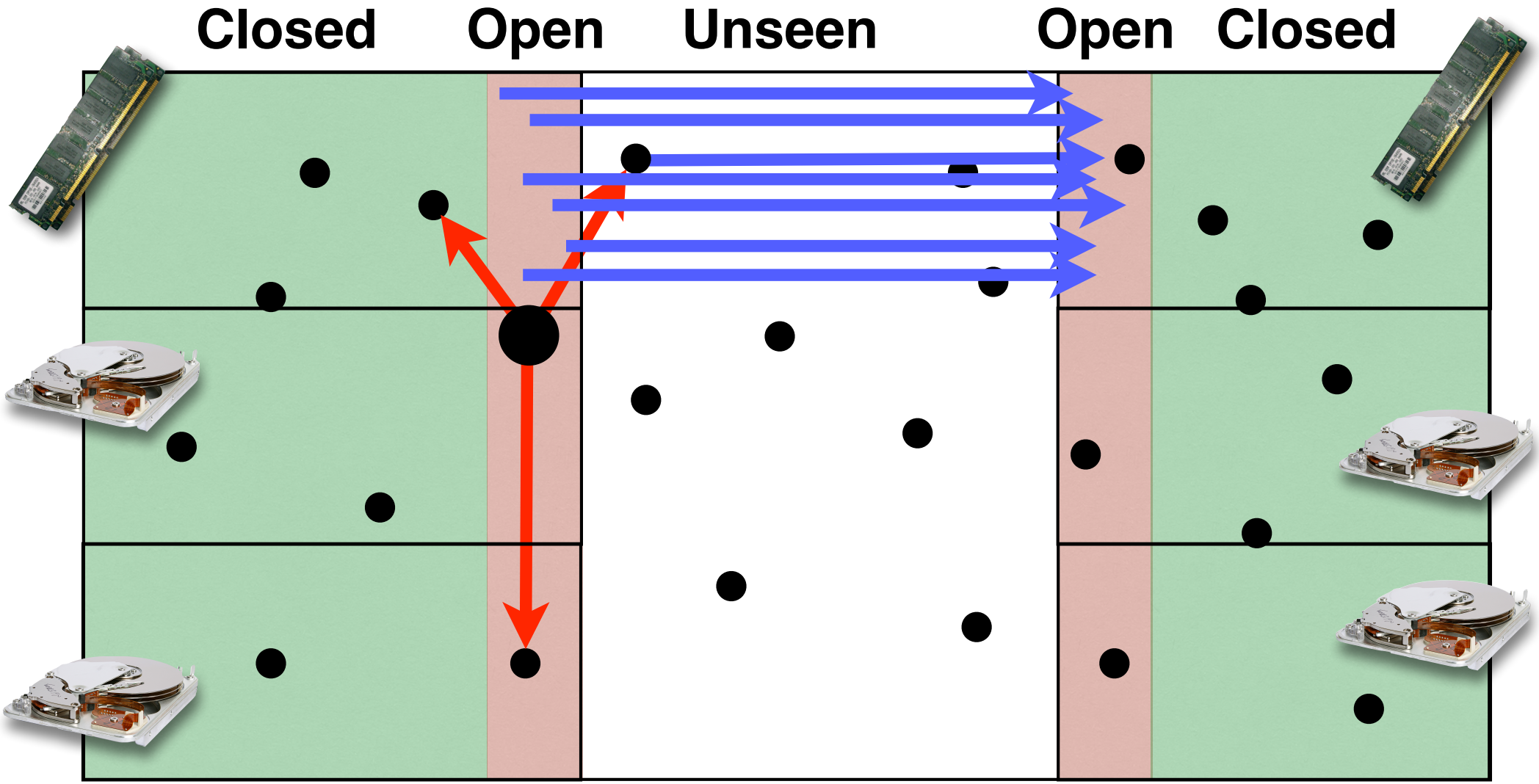
External Memory Bidirectional Search



External Memory Bidirectional Search



External Memory Bidirectional Search



Best-First Search

- Remove from OPEN
  - Check if state is a goal
  - Generate successors
    - Lookup successors in OPEN
    - Lookup successors in CLOSED
  - Add state to CLOSED

Bidirectional Search

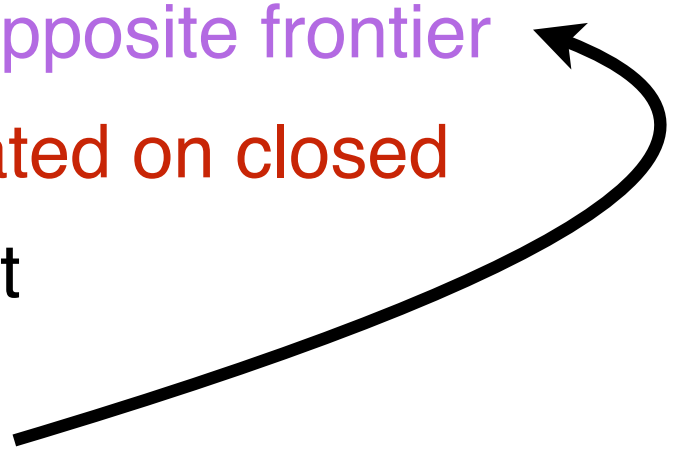
- Remove from OPEN
  - Generate successors
    - Lookup successors in OPEN
    - Lookup successors in CLOSED
    - Check if state is on path to goal
  - Add state to CLOSED

Delayed Solution Detection

- Load best bucket into memory (removed dups)
  - Check bucket against opposite frontier
  - Check if state is duplicated on closed
  - For each state in bucket
    - Generate successors
    - Add successors to OPEN

Delayed Solution Detection

- Load best bucket into memory (removed dups)
  - Check bucket against opposite frontier
  - Check if state is duplicated on closed
  - For each state in bucket
    - Generate successors
    - Add successors to OPEN



PEMM

- Parallel External-Memory Meet in the Middle
  - Delayed Solution Detection
  - Delayed Duplicate Detection
  - Parallel loading, expansion, and solution detection

Results Excerpt
(Rubik's Cube - no heuristic)

Depth	% Exp	%I/O	%DSD
18	38.42	61.58	55.00
18	56.69	43.31	17.33
20	33.04	66.96	56.97

133GB - 5TB of disk

Results Excerpt
(Rubik's Cube - no heuristic)

Depth	% Exp	%I/O	%DSD
18	38.42	61.58	55.00
18	56.69	43.31	17.33
20	33.04	66.96	56.97

133GB - 5TB of disk

Results Excerpt
(Rubik's Cube - no heuristic)

Depth	% Exp	%I/O	%DSD
18	38.42	61.58	55.00
18	56.69	43.31	17.33
20	33.04	66.96	56.97

133GB - 5TB of disk

IDA* Comparison (nodes)

Depth	PEMM none	PEMM 4.8GB	IDA* 80MB	IDA* 4.8GB	IDA* 114GB
18	12.23B	**2.90B**	70.31B	4.78B	**1.01B**
20	38.08B	**38.08B**	-	116B	**24.59B**

IDA* Comparison (time)

Depth	PEMM none	PEMM 4.8GB	IDA* 80MB	IDA* 4.8GB	IDA* 114GB
18	22,643	**16,170**	108,559	**8,999**	**1,652**
20	**100,816**	321,827	-	215,800	**40,048**

Future Work

- Improve solution detection
- Improved termination conditions
- Improved tie-breaking

Acknowledgements

- Work supported by NSF Grant 1551406
- Robert C. Holte, Ariel Felner, Guni Sharon

Extra Results From Paper

Results - Rubik's Cube

(No Heuristic)

Depth	Time (s)	Nodes (B)	% Exp	%I/O	%DSD
18	6,386	2.88	38.42	61.58	55.00
18	22,643	12.23	56.69	43.31	17.33
20	100,816	38.08	33.04	66.96	56.97

133GB - 5TB of disk

Results - Rubik's Cube

(No Heuristic)

Depth	Time (s)	Nodes (B)	% Exp	%I/O	%DSD
18	6,386	2.88	38.42	61.58	55.00
18	22,643	12.23	56.69	43.31	17.33
20	100,816	38.08	33.04	66.96	56.97

133GB - 5TB of disk

Results - Rubik's Cube

(No Heuristic)

Depth	Time (s)	Nodes (B)	% Exp	%I/O	%DSD
18	6,386	2.88	38.42	61.58	55.00
18	22,643	12.23	56.69	43.31	17.33
20	100,816	38.08	33.04	66.96	56.97

133GB - 5TB of disk

Results - Rubik's Cube

(No Heuristic)

Depth	Time (s)	Nodes (B)	% Exp	%I/O	%DSD
18	6,386	2.88	38.42	61.58	55.00
18	22,643	12.23	56.69	43.31	17.33
20	100,816	38.08	33.04	66.96	56.97

133GB - 5TB of disk

Results - Rubik's Cube

(No Heuristic)

Depth	Time (s)	Nodes (B)	% Exp	%I/O	%DSD
16	1,063	1.00	78.19	21.81	12.28
17	3,683	2.13	45.93	54.07	47.12
17	6,031	2.78	36.47	63.53	58.86
17	3,262	2.02	48.32	51.68	44.03
18	11,681	5.77	49.70	50.30	36.06
18	8,245	3.69	40.33	59.67	49.37
18	8,031	3.85	44.19	55.81	43.57
18	8,276	3.98	45.78	54.22	42.85
18	6,386	2.88	38.42	61.58	55.00
18	22,643	12.23	56.69	43.31	17.33
20	100,816	28.08	33.04	66.96	56.97

133GB - 5TB of disk

Results - Rubik's Cube

(No Heuristic)

Depth	Time (s)	Nodes (B)	% Exp	%I/O	%DSD
16	1,063	1.00	78.19	21.81	12.28
17	3,683	2.13	45.93	54.07	47.12
17	6,031	2.78	36.47	63.53	58.86
17	3,262	2.02	48.32	51.68	44.03
18	11,681	5.77	49.70	50.30	36.06
18	8,245	3.69	40.33	59.67	49.37
18	8,031	3.85	44.19	55.81	43.57
18	8,276	3.98	45.78	54.22	42.85
18	6,386	2.88	38.42	61.58	55.00
18	22,643	12.23	56.69	43.31	17.33
20	100,816	28.08	33.04	66.96	56.97

133GB - 5TB of disk

Results - Rubik's Cube

(No Heuristic)

Depth	Time (s)	Nodes (B)	% Exp	%I/O	%DSD
16	1,063	1.00	78.19	21.81	12.28
17	3,683	2.13	45.93	54.07	47.12
17	6,031	2.78	36.47	63.53	58.86
17	3,262	2.02	48.32	51.68	44.03
18	11,681	5.77	49.70	50.30	36.06
18	8,245	3.69	40.33	59.67	49.37
18	8,031	3.85	44.19	55.81	43.57
18	8,276	3.98	45.78	54.22	42.85
18	6,386	2.88	38.42	61.58	55.00
18	22,643	12.23	56.69	43.31	17.33
20	100,816	28.08	33.04	66.96	56.97

133GB - 5TB of disk

Results - Rubik's Cube

(No Heuristic)

Depth	Time (s)	Nodes (B)	% Exp	%I/O	%DSD
16	1,063	1.00	78.19	21.81	12.28
17	3,683	2.13	45.93	54.07	47.12
17	6,031	2.78	36.47	63.53	58.86
17	3,262	2.02	48.32	51.68	44.03
18	11,681	5.77	49.70	50.30	36.06
18	8,245	3.69	40.33	59.67	49.37
18	8,031	3.85	44.19	55.81	43.57
18	8,276	3.98	45.78	54.22	42.85
18	6,386	2.88	38.42	61.58	55.00
18	22,643	12.23	56.69	43.31	17.33
20	100,816	28.08	33.04	66.96	56.97

133GB - 5TB of disk

Results - Rubik's Cube

(No Heuristic)

Depth	Time (s)	Nodes (B)	% Exp	%I/O	%DSD
16	1,063	1.00	78.19	21.81	12.28
17	3,683	2.13	45.93	54.07	47.12
17	6,031	2.78	36.47	63.53	58.86
17	3,262	2.02	48.32	51.68	44.03
18	11,681	5.77	49.70	50.30	36.06
18	8,245	3.69	40.33	59.67	49.37
18	8,031	3.85	44.19	55.81	43.57
18	8,276	3.98	45.78	54.22	42.85
18	6,386	2.88	38.42	61.58	55.00
18	22,643	12.23	56.69	43.31	17.33
20	100,816	28.08	33.04	66.96	56.97

133GB - 5TB of disk