

# Hot/Cold Data Identification

Suggestion

Yoo Kiseong

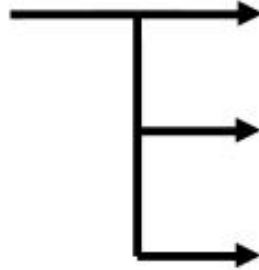


# INDEX

- **Data Structure**
- **Operations**
- **Frequency Identification**
- **Recency Identification**
- **Deploying 'Stack Distance'**

# Data Structure

Double Linked List



Node	Node	Node	

Seperate List

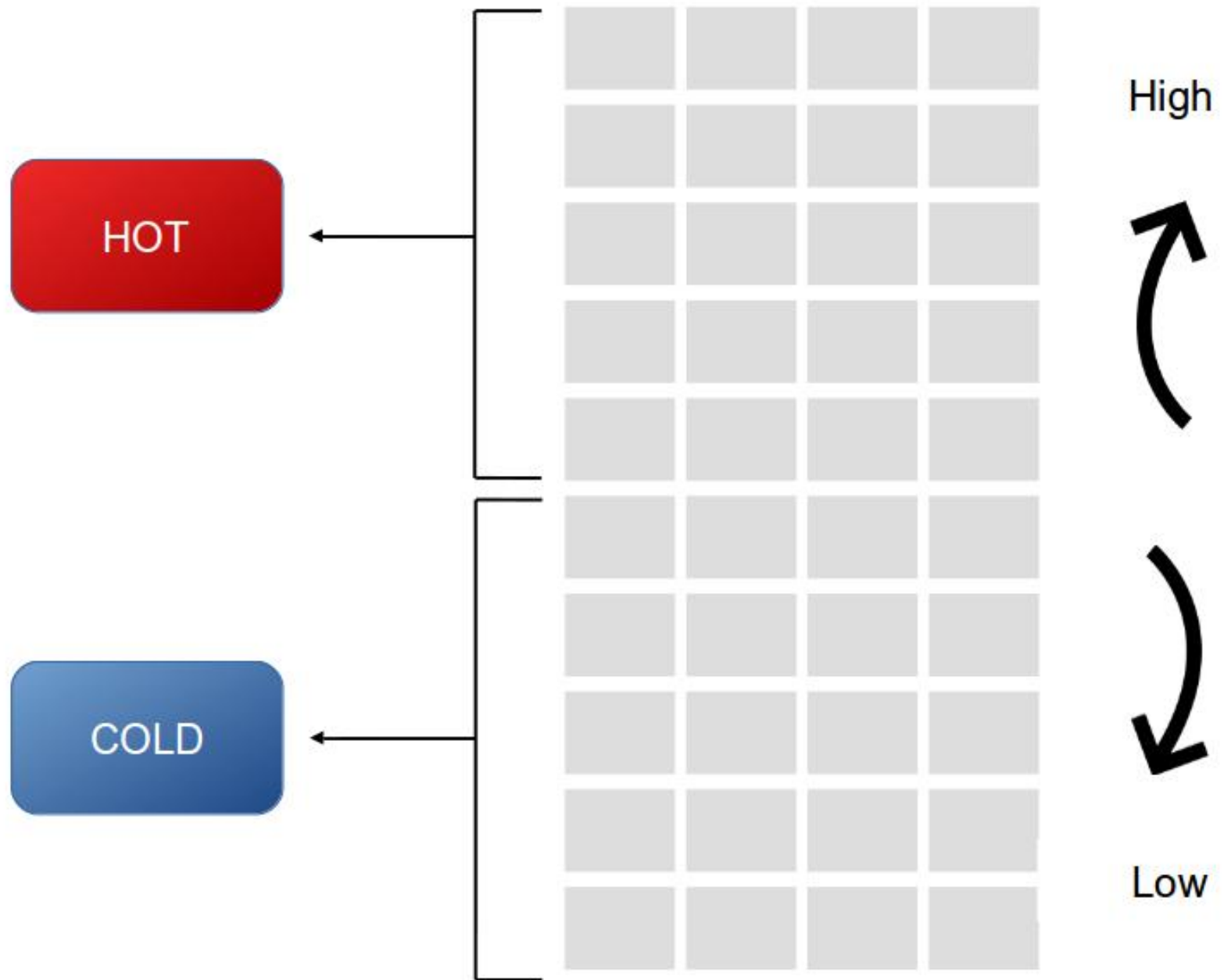
Seperate List

Seperate List

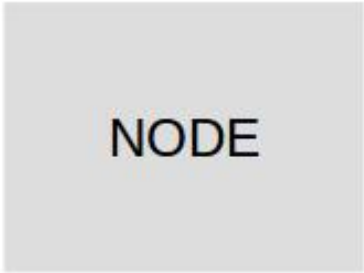
\* Each List is  
seperate from lists above and below

\* Each List has  
limit for maximum size

# Data Structure



# Data Structure



NODE

- int jump : how many levels this jumped previously
- address\_t\* blk\_address : pointer to the block  
or a hash value of the block with smaller digits than address itself
- etc.

# Operations

- Insert

- Remove

- Jump

  - Heat (special case of jump operation)

- Push

  - Down (special case of push operation)

# Operations – 1) Insert

	A	B	C	D
1				
2				
3				
4				
5				
6	A			
7				
8				
9				
10				

# Operations – 1) Insert

	A	B	C	D
1				
2				
3				
4				
5				
6	B	A		
7				
8				
9				
10				



# Operations – 1) Insert

	A	B	C	D
1				
2				
3				
4				
5				
6	C	B	A	
7				
8				
9				
10				

# Operations – 1) Insert

	A	B	C	D
1				
2				
3				
4				
5				
6	D	C	B	A
7				
8				
9				
10				

# Operations – 3) Jump

1				
2				
3				
4				
5				
6	D	C	B	A
7				
8				
9				
10				



# Operations – 3) Jump

ADD BAD

1				
2				
3				
4				
5	A			
6	D	C	B	
7				
8				
9				
10				



# Operations – 3) Jump

A D D B A D

1				
2				
3				
4				
5	D	A		
6	C	B		
7				
8				
9				
10				



# Operations – 3) Jump

ADD BAD

1				
2				
3	D			
4				
5	A			
6	C	B		
7				
8				
9				
10				



# Operations – 3) Jump

ADD **B** AD

1				
2				
3	D			
4				
5	B	A		
6	C			
7				
8				
9				
10				



# Operations – 3) Jump

A D D B A D

1				
2				
3	A	D		
4				
5	B			
6	C			
7				
8				
9				
10				

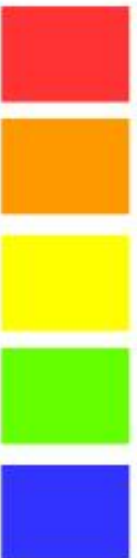




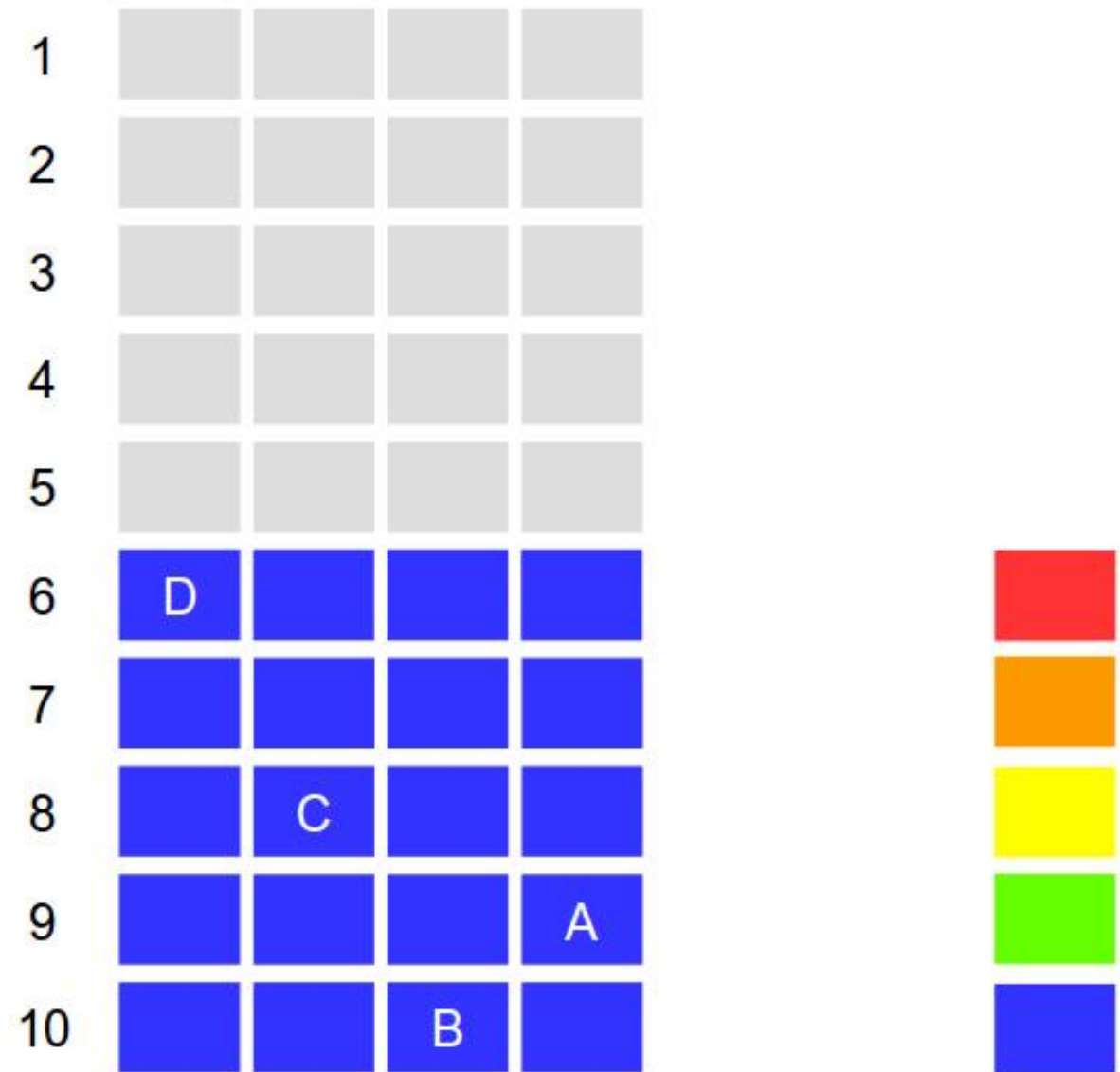
# Operations – 3) Jump

A D D B A D

1	D			
2				
3	A			
4				
5	B			
6	C			
7				
8				
9				
10				



# Operations – 3-1) Heat



# Operations – 3-1) Heat

A D A

1				
2				
3				
4				
5				
6	A	D		
7				
8			C	
9				
10				B



# Operations – 3-1) Heat

A D A

1				
2				
3				
4				
5				
6	D	A		
7				
8			C	
9				
10				B



# Operations – 3-1) Heat

A D A

1				
2				
3				
4				
5	A			
6	D			
7				
8			C	
9				
10				B



# Operations – 4) Push

1				
2				
3				
4				
5				
6	D	C	B	A
7				
8				
9				
10				



# Operations – 4) Push

E

1				
2				
3				
4				
5				
6	E	D	C	B
7	A			
8				
9				
10				



# Operations – 4) Push

1	D	C	B	A
2				
3				
4				
5				
6				
7				
8				
9				
10				





# Operations – 4) Push

E

1	E	D	C	B
2	A			
3				
4				
5				
6				
7				
8				
9				
10				



# Operations – 4-1) Down

1				
2				
3				
4				
5	D	C	B	A
6	E		F	
7				
8				
9				
10				



# Operations – 4-1) Down

E F

1				
2				
3				
4				
5	E	D	C	B
6	A		F	
7				
8				
9				
10				



# Operations – 4-1) Down

E F

1				
2				
3				
4				
5	F	E	D	C
6	B	A		
7				
8				
9				
10				



# Operations – 2) Remove

1				
2				
3				
4				
5				
6				
7				
8				
9				
10			Y	X



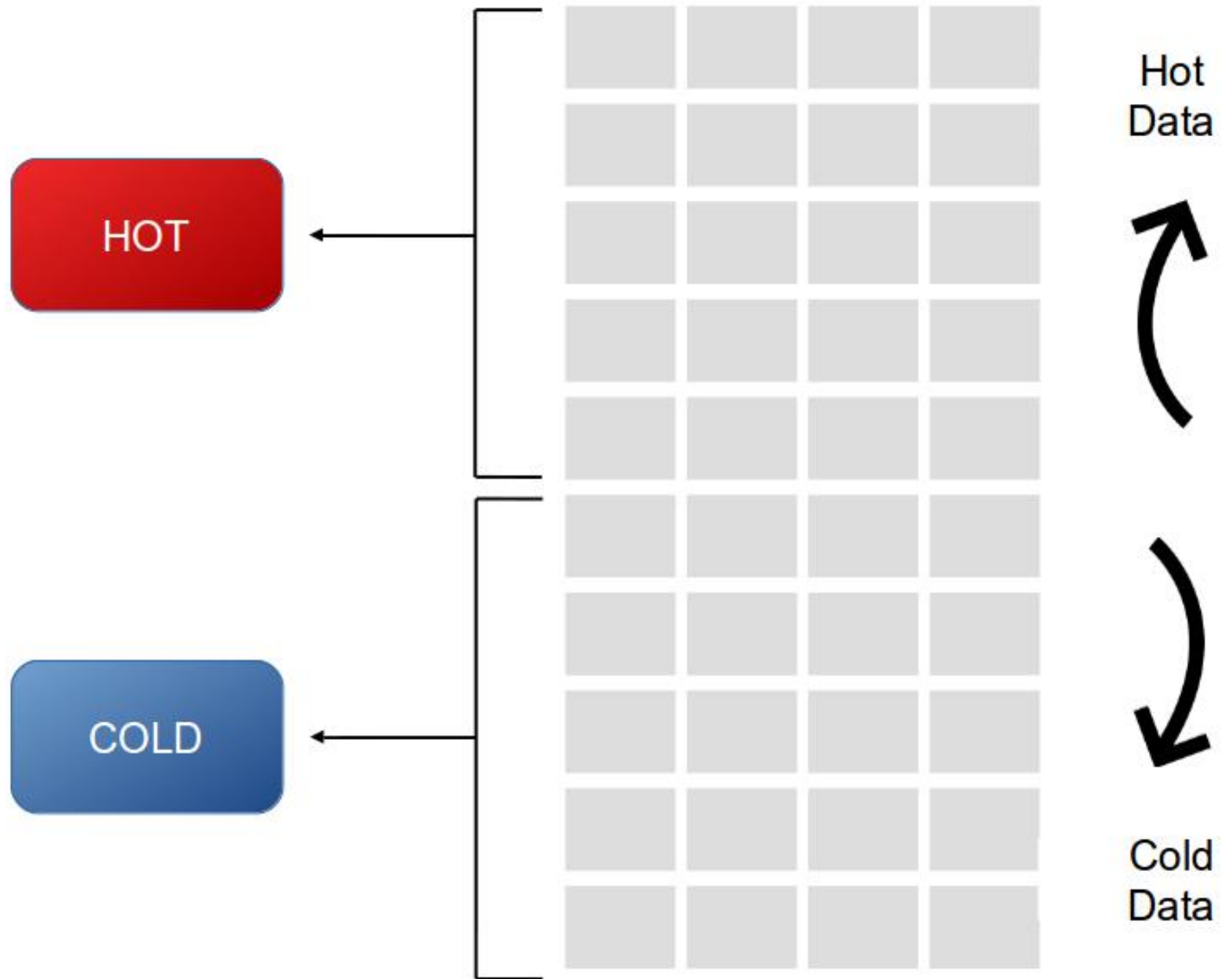
# Operations – 2) Remove

blk

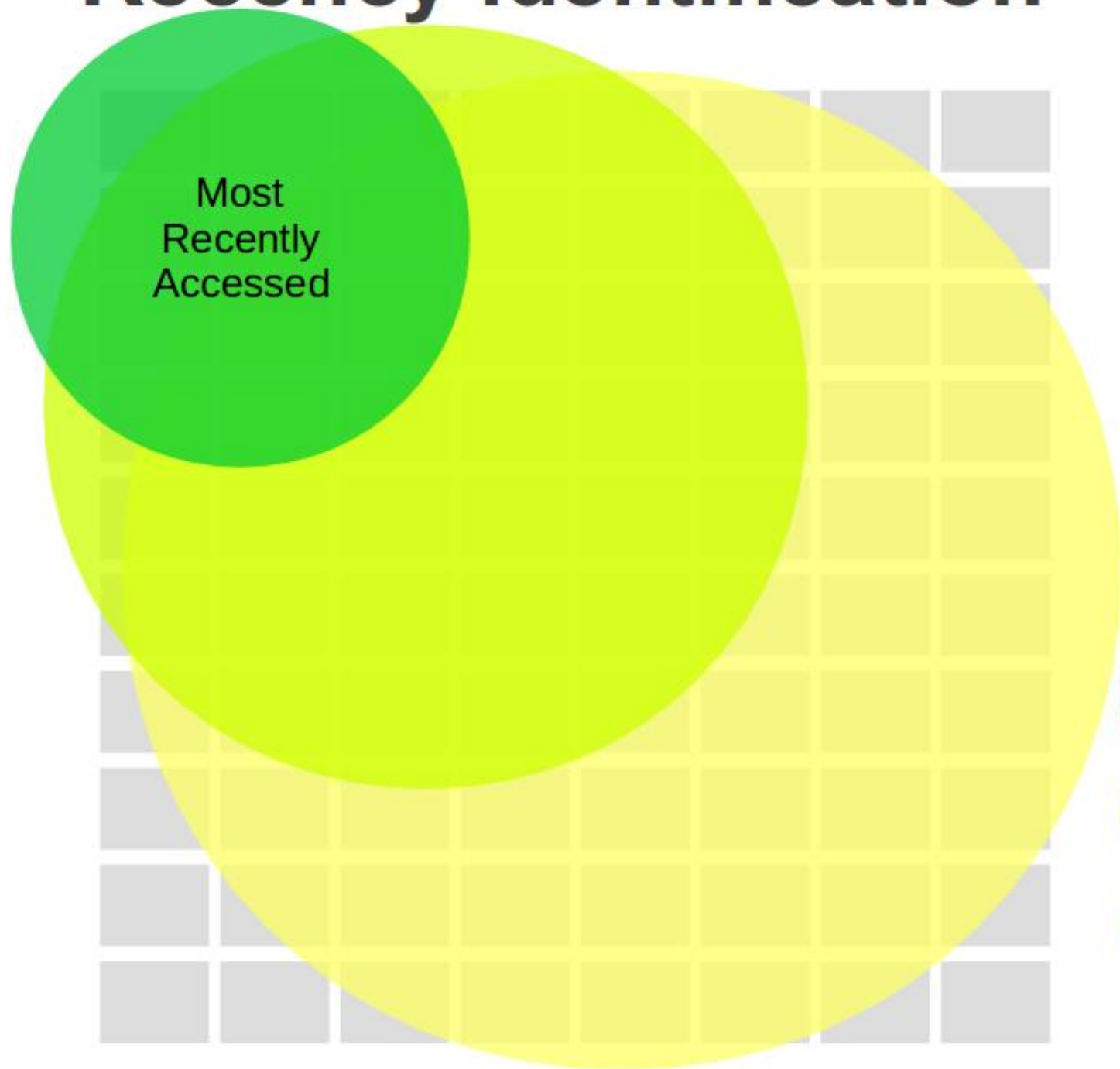
1				
2				
3				
4				
5				
6	blk			
7				
8				
9				
10				Y



# Frequency identification



# Recency identification



Most  
Recently  
Accessed




# Deploying 'Stack Distance'

Idea 1

=> Variable Maximum Size of Each List

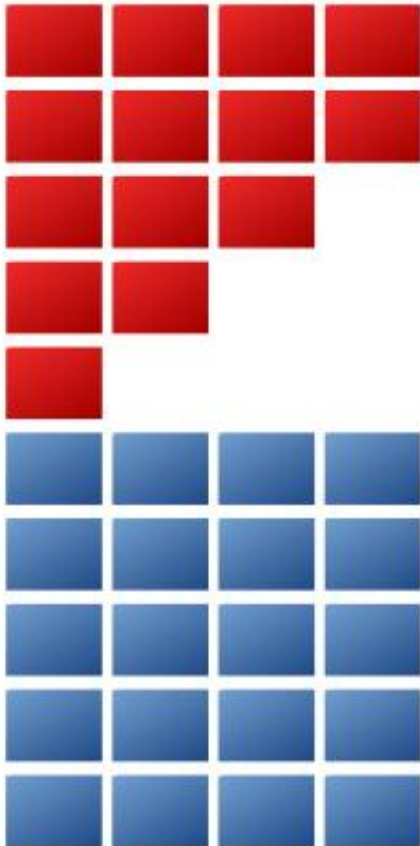
Idea 2

=> ...



# Deploying 'Stack Distance'

If (stack distance > threshold) then :  
decrease HOT area sz\_limit





**Thank you**