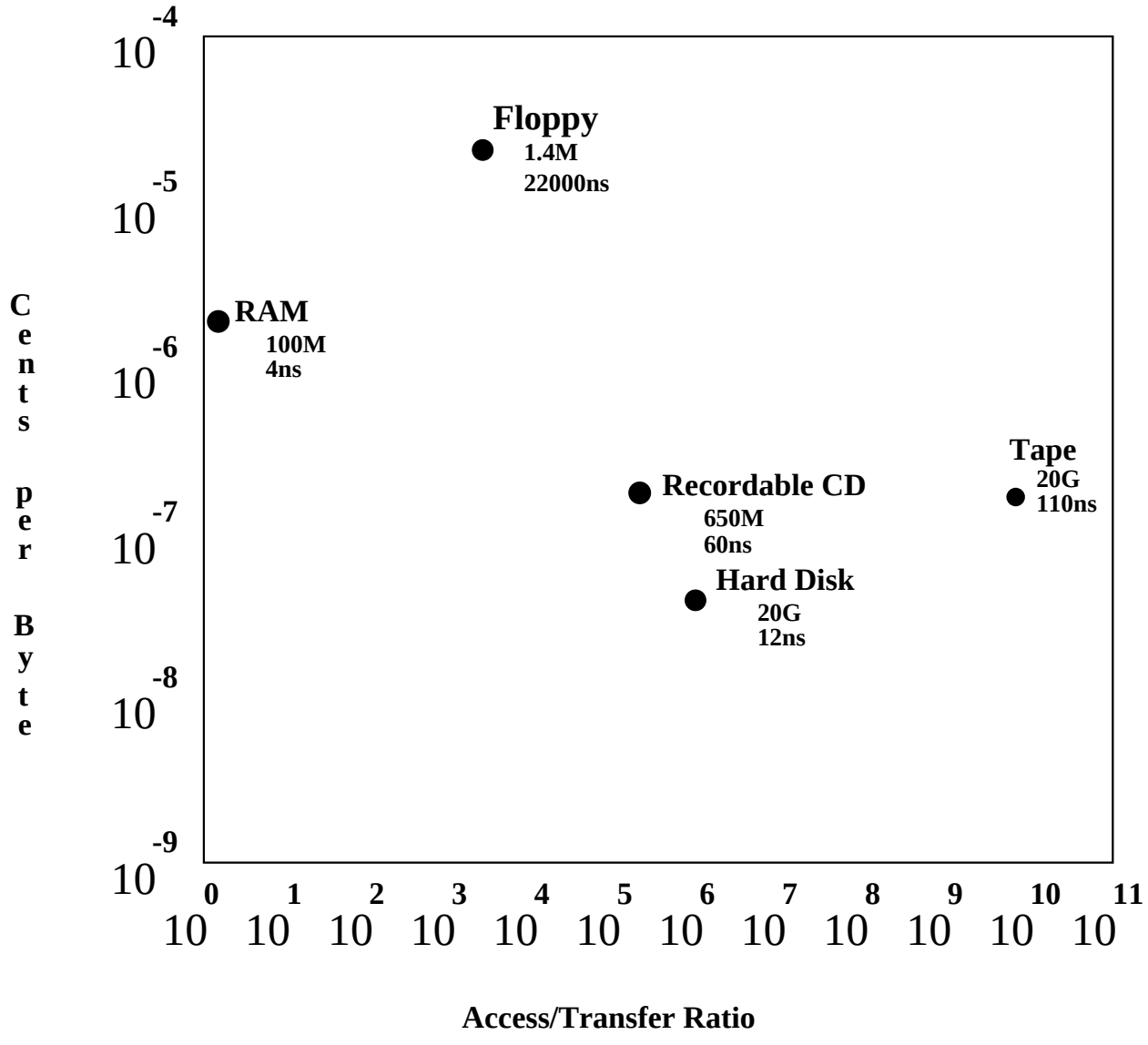
The author gratefully acknowledges support from the taxpayers of Québec and of Canada who have paid his salary and research grants while this work was developed at McGill University, and from his students (who built the implementations and investigated the data structures and algorithms) and their funding agencies.

T. H. Merrett ©98/9

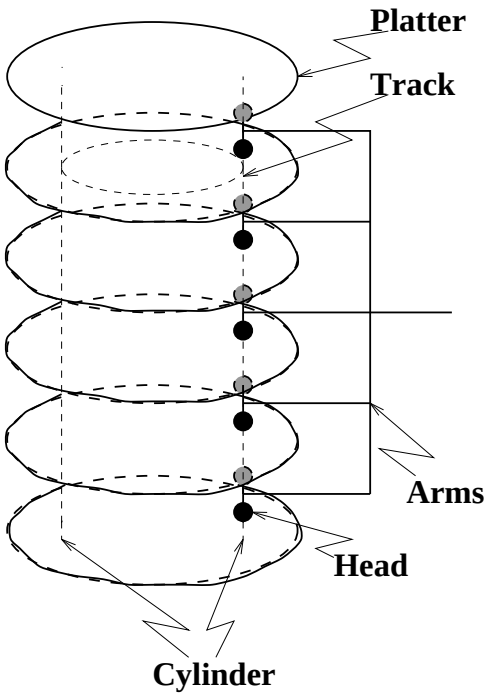
Exhibit 1.2.1 Cost vs. Access/Transfer Ratio for 1998 Memory Technology, Showing Typical Unit Capacities and Transfer Times per Byte.



The Difference Between RAM and SS

e.g., Disks

- have been around a long time
- recently improved faster than other technologies



  - Latency @ 7200 rpm: $\lambda = 1/(2 \times 120)$ ~ 4 ms.
  - Arm movement, say, $\mu = 8$ ms.

- Access time $\lambda + \mu = 12$ms.
- With, say, $\beta = 100,000$ bytes/track, transfer is $1/\tau = 12$Mbytes/sec

So access/transfer $\rho = (\lambda+\mu)/\tau = 144,000$ (bytes which could have been transferred while seeking the data).

Exhibit 1.2.2 Specifications for Magnetic Disk Units to be Used in This Book (revised 1998).

	τ TRANSFER TIME/BYTE	ρ ACCESS/ TRANSFER RATIO	σ ROTATION SPEED
DISK2000	8.3nsec.	1,440,000	7200 rpm
RCD2000	0.7μsec.	240,000	600–300 rpm
FLOPPY2000	22μsec.	4,000	600 rpm

	λ AVERAGE LATENCY	μ AVERAGE ARM DELAY	β BYTES/ TRACK
DISK2000	4.2ms	7.8ms	1,000,000
RCD2000	75ms	95ms	140,000–280,000
FLOPPY2000	50ms	38ms	4608

	γ TRACKS/ CYLINDER	ν CYLINDERS /UNIT	TOTAL CAPACITY
DISK200	10	20000	20GB
RCD2000	1	3095	650MB
FLOPPY2000	2	160	1.4MB

	τ TRANSFER TIME/BYTE	ρ ACCESS/ TRANSFER	σ TAPE SPEED
TAPE2000	12.5nsec.	20 G	80 ips

	λ AVERAGE LATENCY	REWIND TIME	δ RECORDING DENSITY
TAPE2000	25 sec.	25sec.	1MBpi

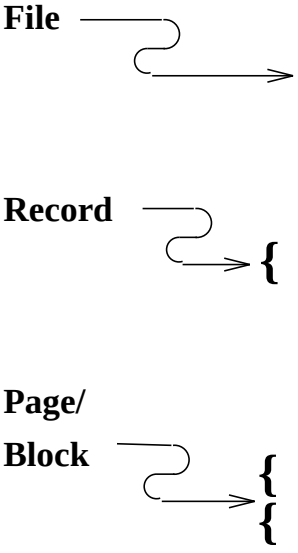
	ι INTER- BLOCK GAP	ϕ TAPE LENGTH	TOTAL CAPACITY
TAPE2000	0.2 in.	3250 ft.	40 GB

Course Overview

- Access Complexity
  - Sequential (linear)
  - (Tree) Logarithmic
  - Direct (constant)
- Activity (*ratio, records needed / total records*)
- Volatility (*add, delete, change records*)
- Symmetry (*all fields are equal?*)

Sequential Files

Ordered



Ord#	*Cust*	*Sales*
1	GNS	E
2	NYC	N
3	L&S	E
4	PR	H
5	NYC	H
6	B&O	H
7	GTRC	N
8	GNS	E

$N =$ 8 records

$n = 4$ blocks

blocksize $b = 2$

load factor $\alpha = 1$

$n = \lceil \frac{N}{b\alpha} \rceil$

Unordered

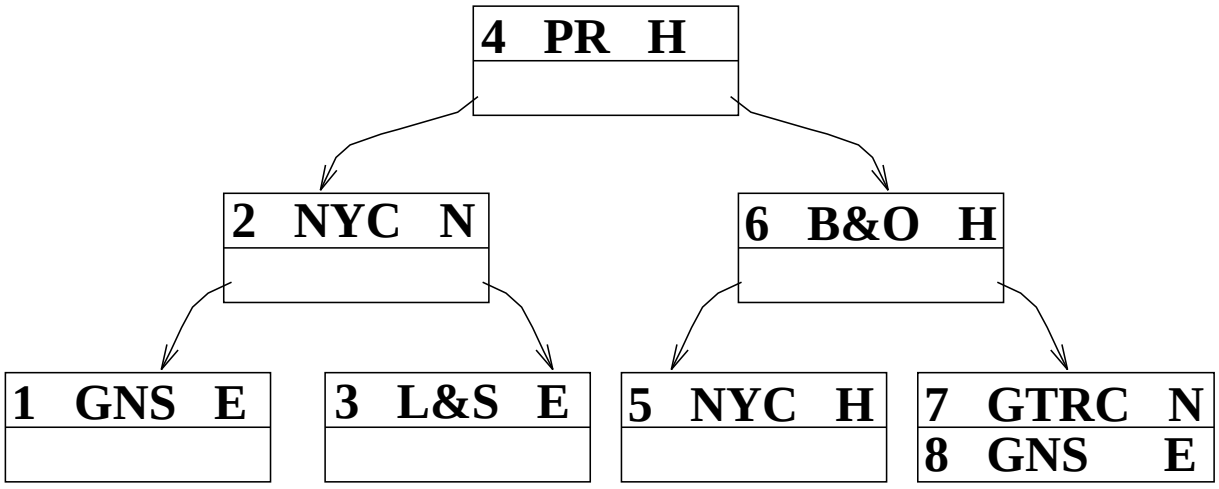
Average cost of a successful search: $n/2$ accesses.

Sequential files are best for *high activity*.

i.e. $>\sim 0.1\%$ of records accessed.

Logarithmic Files

e.g., B-trees



Average cost of a successful search:
log n accesses.

e.g., $n = 6$ $\lceil \log_2 n \rceil = 3$

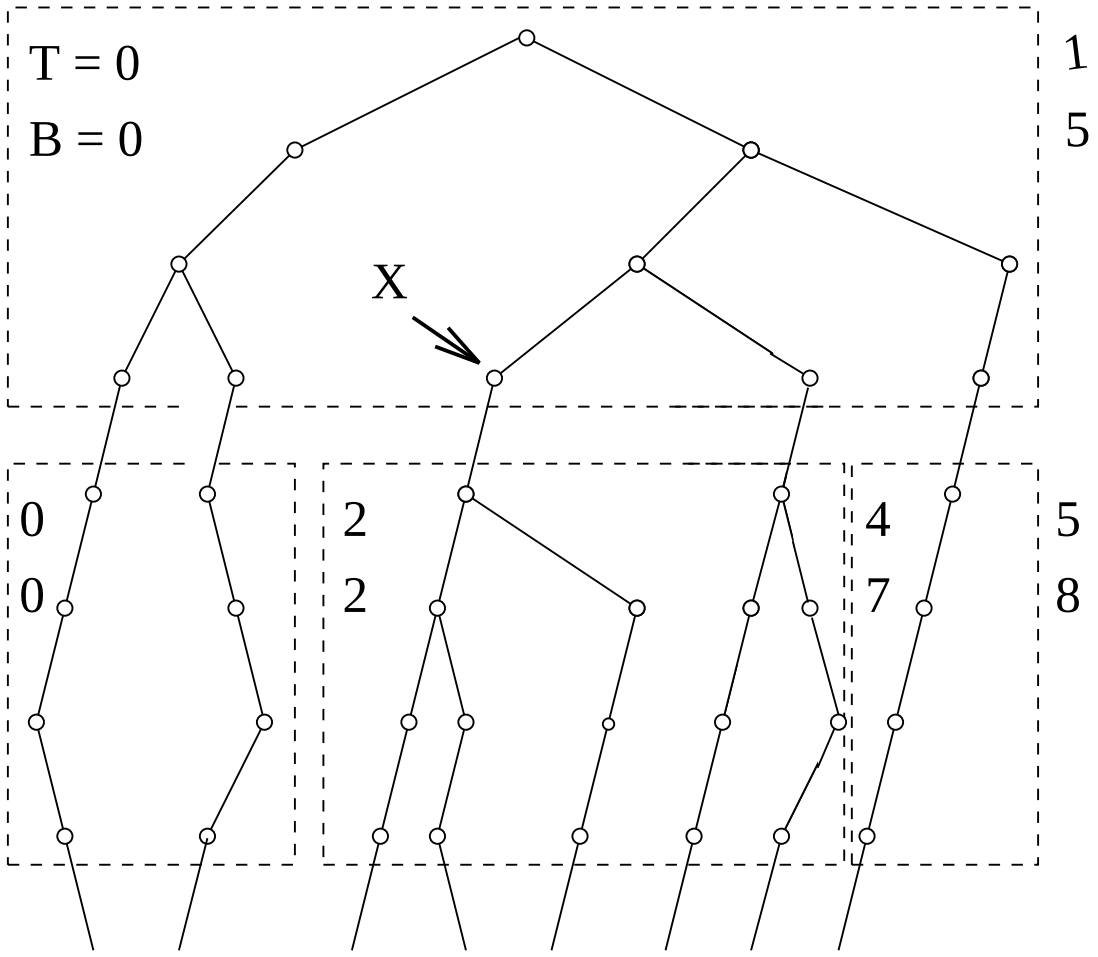
B-trees are very flexible, good for *dynamic data*

Tries

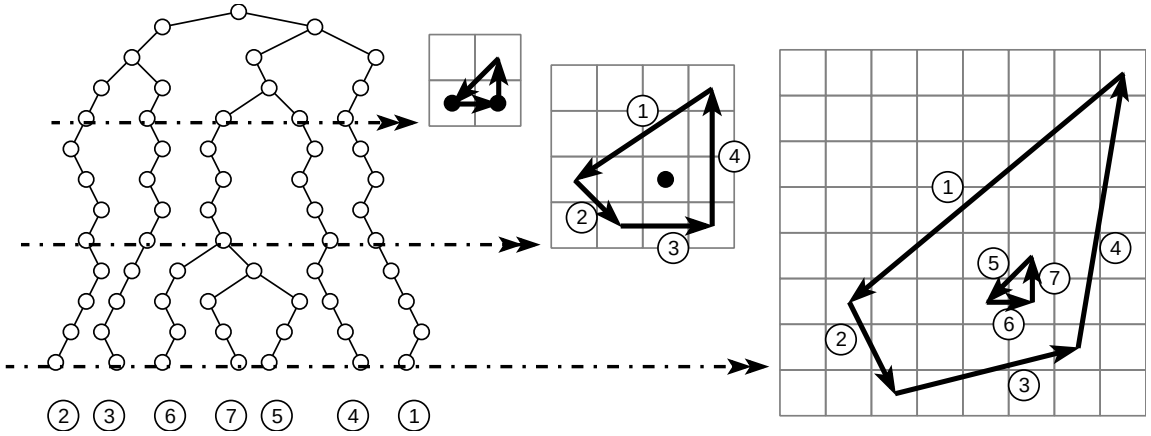
(Digital trees Information re*trie*val)

Sample data:

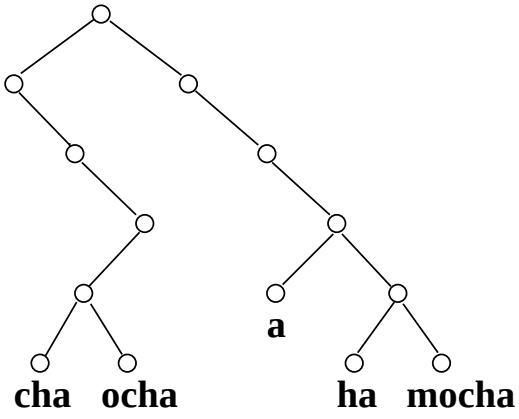
00000011
00101100
10000000
10000101
10001000
10100000
10101100
11000000

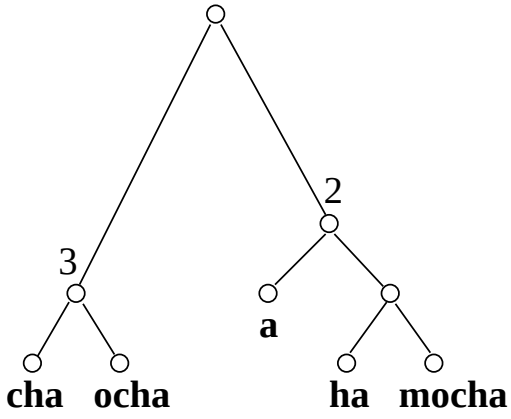


Kd-Tries and Variable Resolution



Truncated Tries and Text Data





1) Truncated Trie

2) PATRICIA Trie

Sample "text":

`mocha :  11101101011011110110001111101000111000 01`

with "starts" every eight bits.

Direct Access Files

e.g., Multipaging

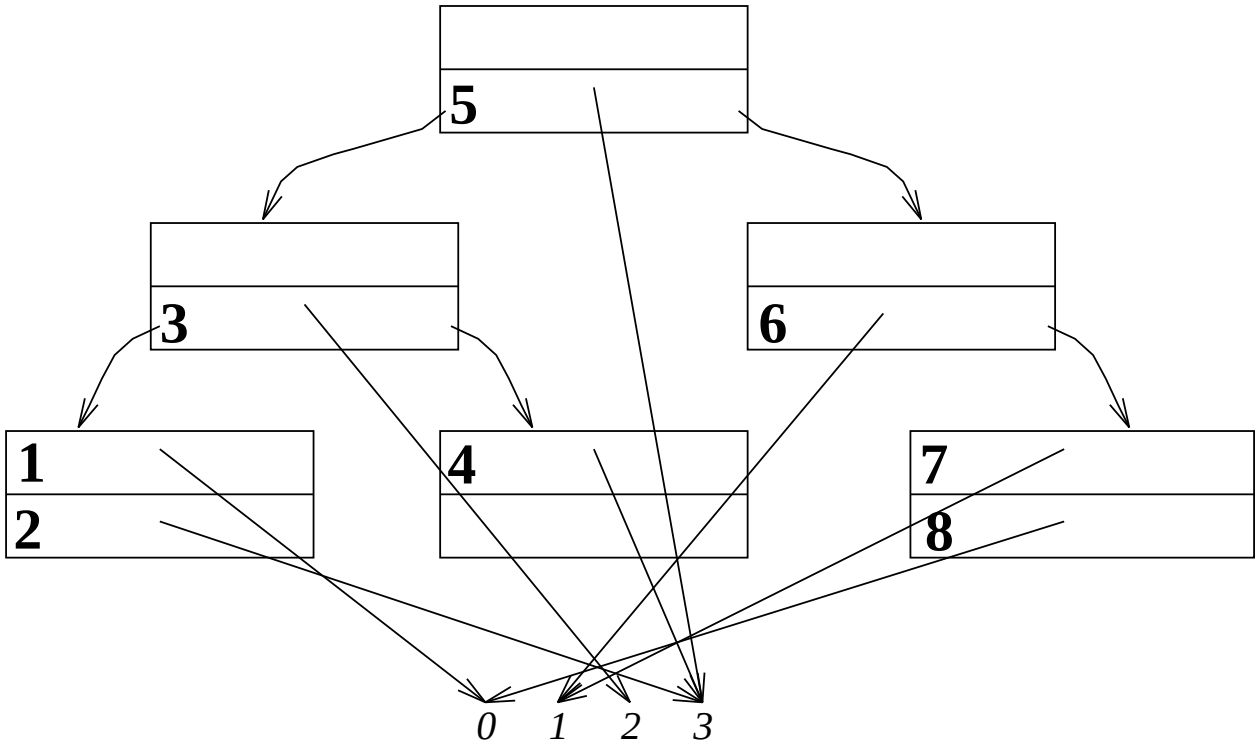
Average cost of a successful search:
1 access.

Order-preserving,
thus good for high activity.

Can be built up dynamically.

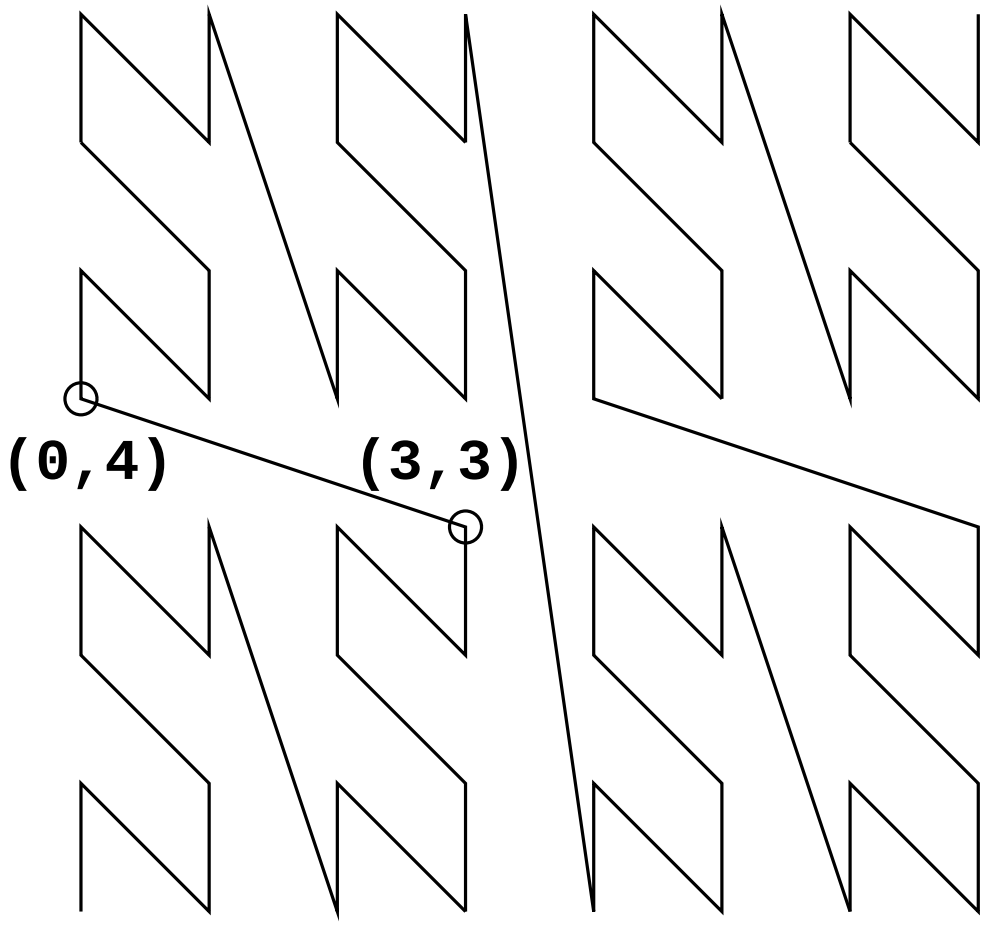
B&O		6	
GNS	1,8		
GTRC			7
L&S	3		
NYC		5	2
PR		4	
	E	H	N

Hybrid Files



B&O GNS GTRC	0 1,8	6 1 7
L&S NYC PR	3 2	3 5 2 4
	E	H N

Z-Ordering



1-dimensional ordering of m-dimensional data

So can use existing structures (e.g., B-tree)

Based on *kd-trie*, or on interleaving of bits:

(3,3)=(0011,0011) shuffles to 0000111 $<$
00010000 unshuffles to (0000,0100)=(0,4)