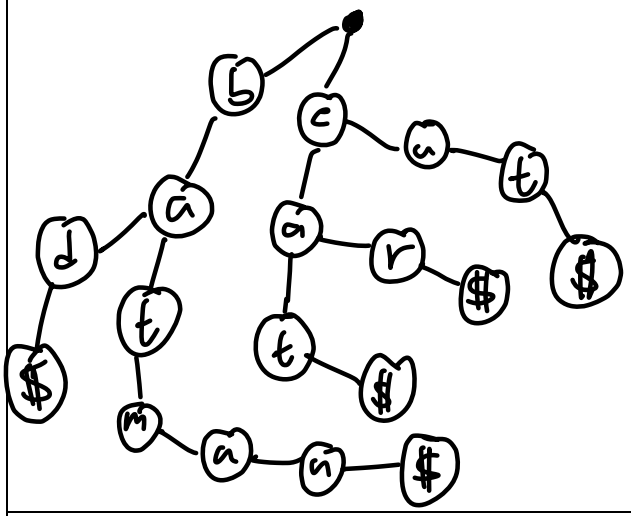
Trie

$ ← termination symbol

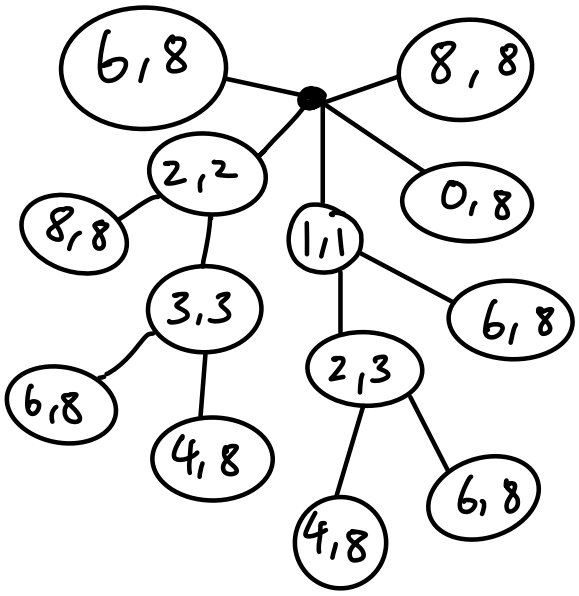
{cat, bat, bad, batman, car, cut}



Suffix Trie

- Mainly used for efficient substring search
- Can be compressed to save space (using only index)

ID	0	1	2	3	4	5	6	7	8
0	w	o	|	o	|	o	a	|	$
1	o	|	o	|	o	a	|	$
2	|	o	|	o	a	|	$	
3	o	|	o	a	|	$		
4	|	o	a	|	$				
5	o	a	|	$					
6	a	|	$						
7	|	$							
8	$								



① String Compression

② indexes instead of string

Suffix array

Suffix Array

- Construction

ID											
1	K	O	K	O	K	R	U	N	C	H	$
2	O	K	O	K	R	U	N	C	H	$	
3	K	O	K	R	U	N	C	H	$		
4	O	K	R	U	N	C	H	$			
5	K	R	U	N	C	H	$				
6	R	U	N	C	H	$					
7	U	N	C	H	$						
8	N	C	H	$							
9	C	H	$								
10	H	$									
11	$										

sort →

ID	
$	11
C	9
H	10
K	1
K	3
K	5
N	8
O	2
O	4
R	6
U	7

Prefix Doubling

ID	
1	K
2	O
3	K
4	O
5	K
6	R
7	U
8	N
9	C
10	H
11	$

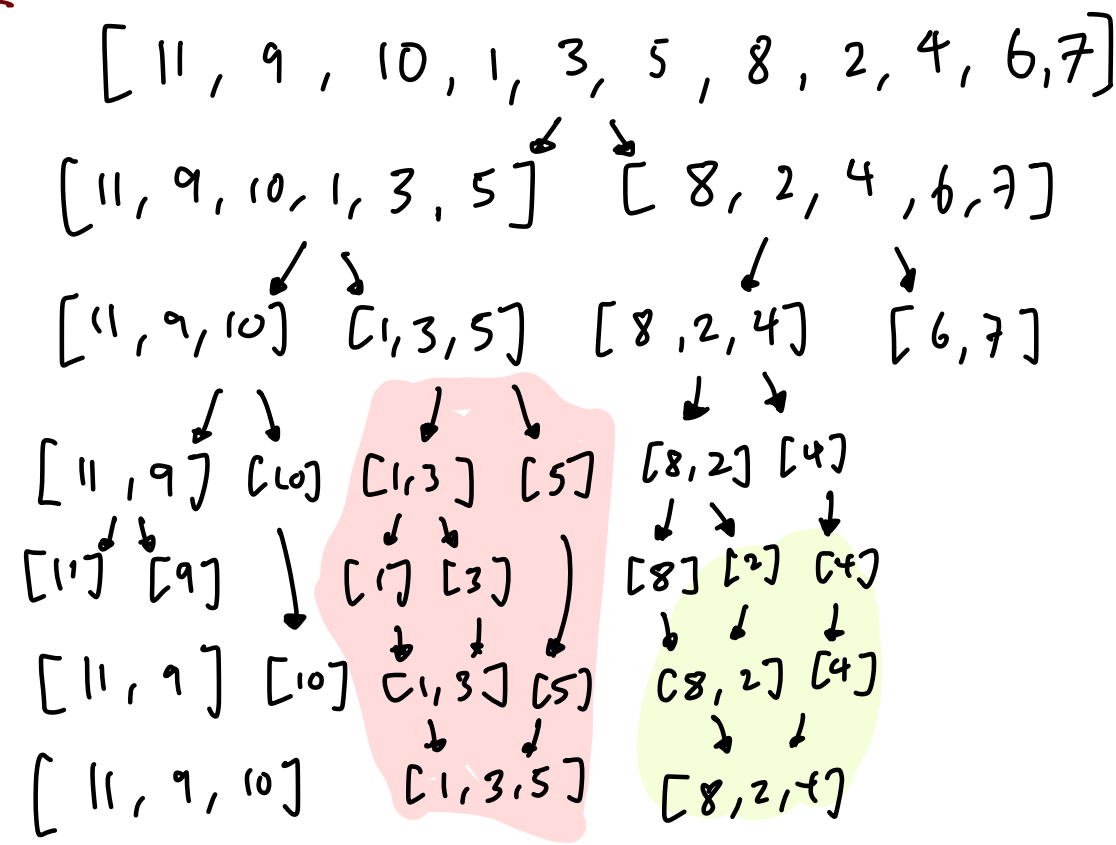
Count Sort →

ID		rank table
$	11	1
C	9	2
H	10	3
K	1	4
K	3	4
K	5	4
N	8	7
O	2	8
O	4	8
R	6	10
U	7	11

k = 1	2	4	8	16

merge sort

	ID	rank table
$	11	1
C	9	2
H	10	3
K	1	4
K	3	4
K	5	4
N	8	7
O	2	8
O	4	8
R	6	10
U	7	11



for each merging / comparison, IF SAME RANK

→ compare based on ranks

ID	+	k	=	Compared rank
1	+	1	=	[2] = 8
3	+	1	=	[4] = 8
5	+	1	=	[6] = 10
2	+	1	=	[3] = 4
4	+	1	=	[5] = 4

#Repeat until all ranks are unique

	ID	rank table
$	11	1
C	9	2
H	10	3
K	1	4
K	3	4
K	5	6
N	8	7
O	2	8
O	4	8
R	6	10
U	7	11

updated rank