

(Radix) Sort Algorithm \rightarrow 1's, 10's, 100's

* Non-Comparison Algorithm
 * Multiple digits
 * Constant length strings.
 P1: 325 042 009 065 002 084
 P2: 002 009 325 042 065 084
 P3: 002 009 042 065 084 325
 * "pavan", "raman", "samal"
 * Find Max 325
 * 3 passes
 * Bucket Sort
 * (0-9) 10 buckets
 Count
 P1 \rightarrow P2 \rightarrow P3 \rightarrow Sort()

How do we control the number of iterations/passes in the radix sort function?

max = 325
 for (int exp = 1; max/exp > 0; exp *= 10)
 {
 countsort(arr, n, exp);
 }
 max = 325
 exp = 1, 10, 100
 325/1 = 325 > 0
 325/10 = 32 > 0
 325/100 = 3 > 0
 325/1000 = 0 < 0

* What is "Hamming Weight"?

11 \rightarrow 1011 \rightarrow 3
 n \rightarrow hamming(int n) {
 11 \rightarrow 5 - 0101
 0 - 0000
 0

n \rightarrow 11 \rightarrow 1011
 1 \rightarrow 0001 & 0001
 0001
 count = 0, 1, 2, 3
 return count;

(n & 1) = 1 \rightarrow 1011 >> 1
 101 \rightarrow 10 \rightarrow 1 \rightarrow 0
 001 \rightarrow 01 \rightarrow 1 \rightarrow 0
 00 \rightarrow 0 \rightarrow 1 stop
 1011 \rightarrow 101 \rightarrow 10 \rightarrow 1 \rightarrow 0
 3

* Max regions in a plane with n straight lines.

* Baker's partition Problem.

Straight Lines	Max Regions	Cuts	Max Pieces
\rightarrow 0	1	0	1
\rightarrow 1	2	1	2
\rightarrow 2	4	2	4
\rightarrow 3	7	3	7
\rightarrow 4	11		
\rightarrow 5	16		

$1, 2, 3, 4, 5 \rightarrow \frac{n(n+1)}{2} = \frac{5 \times 6}{2} = 15$
 $\frac{n(n-1)}{2} + 1$
 $S(n) + 1$

Dynamic Memory Allocation \rightarrow

In case of C Language we use \rightarrow

#include <stdlib.h>

malloc()

calloc()

realloc()

free()

{1, 2, 3}, {4, 5, 6}

{7, 8, 9}

0 1 2
0 1 2
3 4 5
6 7 8
9

$n \times n = 3 \times 3$

In C++

#include <cstdlib>

new ✓

delete ✓

Two pointer Approach
Merge Sort Algorithm \rightarrow

* Merge two sorted arrays

$a_1 = [1, 3, 5, 7, 9]$

$b_1 = [2, 4, 6]$

$a[i] < b[j] \rightarrow i++$

$a[i] > b[j] \rightarrow j++$

$c_1 = [1, 2, 3, 4, 5, 6, 7, 9]$

(m-s+1) (e-m)

4 9 2 1 6 5

$m = \frac{s+e}{2}$

Split

$\log N$

$n \log N$

0 1 2 3 4 5

s LHS m m-s+1 RHS e e-m

4 9 2 1 6 5

4 5 6

1 1 6 5

1 1 6 5

1 1 6 5

1 1 6 5

1 1 6 5

1 1 6 5

1 1 6 5

1 1 6 5

1 1 6 5

1 1 6 5

1 1 6 5

1 1 6 5

1 1 6 5

1 1 6 5

9315 \rightarrow Saurav