Sorting 2

Tuesday, 2 March 2021 8:04 PM

Puzzle:

25 horses, [All different speeds], At any you can choose any 5 horses [$h_1 > h_2 > h_3 > h_4 > h_5$] we get relative order of 5 horse after a race.

→ Find the 1 & 2 & 3 horses overall in minimum possible races.

a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_2	d_2	e_2
a_3	b_3	c_3	d_3	e_3
a_4	b_4	c_4	d_4	e_4
a_5	b_5	c_5	d_5	e_5

$a_1 > b_1 > c_1 > d_1 > e_1$

a_1 is already decided 1st place right

$a_2\ a_3\ b_1\ b_2\ c_1$ → 3rd race

→ Total races = 7 races

$a_1 > b_1 > c_1 > d_1 > e_1$

Not possible

cannot be answer

b_1

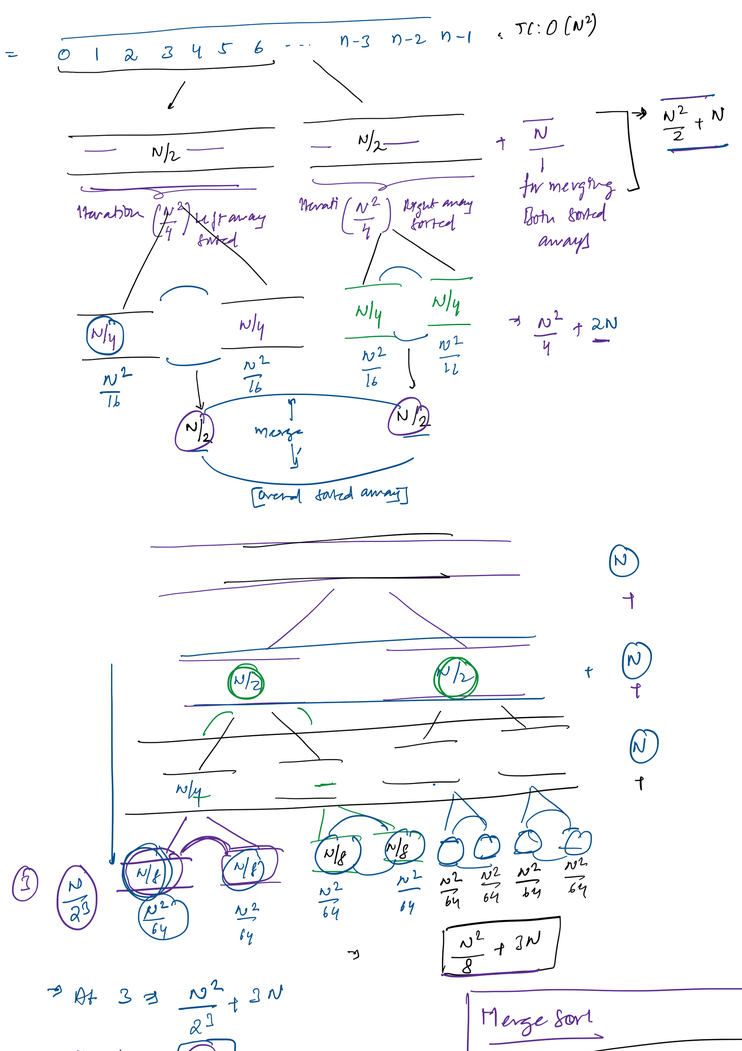
c_1

→ 8 races : [pointer approach]

Merge Sort

ar[10] = 10 3 8 15 6 12 2 18 7 1

ar[n] = 0 1 2 3 4 5 6 ... n-3 n-2 n-1 , TC: $O(N^2)$



→ At 3 → $\frac{N^2}{2^3} + 3N$

→ At k → $\frac{N^2}{2^k} + kN$

Array : $\frac{N}{2^k}$

$\frac{N}{2^k} = 1$, $N = 2^k$ $k = \log_2 N$

Iterations → $\frac{N^2}{2^k} + kN$

→ $\left[\frac{N^2}{2^{\log_2 N}}\right] + (N)(\log_2 N)$

→ $\left[\frac{N^2}{N}\right] + N \log N$

→ $N + N\log N$

Merge Sort

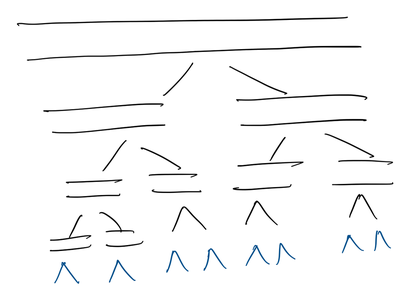
1 2 3 6 7 8 10 12 15 18

3 6 8 10 15 1 2 7 12 18

3 8 10 6 15 2 12 18 1 7

3 10 8 15 2 12 18 7 1

10 3 12 2



// Given an array sort entire array from 0- N-1

	0	1	2	3	4	5	6	7	8	9
ar[]:	3	6	8	10	15	2	12	18	1	7

lo, hi, mi = (lo+hi)/2

0 – 9 4

→ if (lo == hi) // single element

5 – 9 → tmp1[5] = {1 2 7 12 18}

copy actual array from which index to which index → [5 – 9]

2nd point

lo2 hi2 mi2: 0 – 4 2

5 – 7 | 8 – 9

tmp2[5] = {3, 6, 8, 10, 15}

[copy back to actual]

lo3 hi3 mi3: 0 – 2 1 | 3 – 4

tmp3[3] = {3 8 10}

[copy back to actual]

lo4 hi4 mi4: 0 – 1 0 | 2 – 2

0 – 0 | 1 – 1

10 | 3

tmp4[2] = {3, 10}

[copy back to original array]

② ar[]: 10 3 8 15 6 12 2 18 7 1

(i, j) i < j & ar[i] > ar[j]

16 2 4 5 2 3 1 2 1 0 = 26

Inversion count of an array

10 3 8 15 6 | 12 2 18 7 1

array = inversion count = 5 | array inversion = 7 | = 12

// we missed combinations

3 6 8 10 15 | 1 2 7 12 18

→ C = 0

i = 0 → N

j = i+1 → N

if ar[i] > ar[j] C++

→ $O(N^2)$

Max Inversion Count

10 7 6 2 0

4 3 2 1 0] $\frac{(5)(4)}{2}$

→ N Elements

→ $\frac{[N][N-1]}{2}$

Descending order

If Inver count = 0 : Ascending order

If InverCou ≈ 0 : [Almost Sorted]

If Inversion count is more you are farthe from Ascending order

int c = 0; // → Datatype of c: long ?

void MS(int ar[], int lo, int hi){

if(lo == hi) {return}

int mid = (lo+hi)/2

//Left array [lo - mid]

MS(ar, lo, mid)

// right array [mid+1, hi]

MS(ar, mid+1, hi)

merge(ar, lo, mid, high)

}

Main logic completed

1) Assumption : Sort Entire array from lo-hi

2) Mainlogic

3) Base Condition

→ TC : $O(N\log N)$

SC : $O(N)$

Stack space : (logN)

[Number of levels of recursion]

void merge(int ar[], int l, int m, int h)

int tmp[h-l+1];

int p1 = l, p2 = m+1, p3 = 0

while (p1 <= m && p2 <= h){

if(ar[p1] > ar[p2]) { tmp[p3++] = ar[p2++] c = c + [m - p1 + 1]

else {tmp[p3++] = ar[p1++]}

}

while(p1 <= m){ tmp[p3++] = ar[p1++] }

while(p2 <= h){ tmp[p3++] = ar[p2++] } // copy remaining

//copy to original array l to h

for(int i = l; i <= h; i++){

ar[i] = tmp[i-l];

}

}

comparing code is kept in a small function → compare(int, int)

→ you can override this compare function as per your convenience

→ compare(obj x, obj y)

// If you want x to come before y how to write condition

If return T: x y

If return F: y x

n, y : x < y

return x < y :

cross count :

1 2 3 6 7 8 10 12 15 18

+5 +5 0 0 +3 +1 → 14

+c

→ remaining number of ele in left[]

At any point = [p1 mid] → [mid - p1 + 1]

10 3 8 15 6 12 2 18 7 1

→ 5 +2 +3 +2 → 7 → 14 → 26

3 6 8 10 15 | 1 2 7 12 18

+1 | 6 15 +1 | 0 | 1

3 8 10 | 2 12 18 | 1 7 c +1

+1 3 10 | 8 | 15 | 6 | 1 | 2 12 | 18 | 1 7

3 10 | 12 2

Dating Algo:

Movies : 1 3idiots, 2 PK, 3 Dangal, 4 Infinity war, 5 Endgame, 6 Chennai Express, 7 Harry Potter

Aakash : 4 5 7 3 1 2 6

Appoov : 1 2 3 4 7 5 6

Chakravarti : 7 3 4 2 1 5 6

Venkat : 5 4 3 2 1 6 7

Sruti : 3 4 5 7 1 6 2

Tripathy : 3 1 2 6 4 7 5

Sandeep : 1 2 3 4 5 6 7

Akhil : 6 5 7 4 3 2 1 → 21

7 6 5 4 3 2 1

max Inve :

21% → 100% Mismatch

	1	2	3	4	5	6	7
Appoov :	1	2	3	4	7	5	6
Sruti :	3	4	6	5	1	7	2
Tripathy :	3	1	2	7	4	5	6
Sandeep :	1	2	3	4	6	7	5

70% | 0 %

			Match Perc
Inversion	10	(10/21)(100) = 47.61	53%
Inve	5	(5/21)(100) = 23.80	77%
Inve	2	(2/21)(100) = 9.5%	91%