

②

index	0	1	2	3	4	5	6
value	17	2	3	8	5	11	1

a) Merge Sort

Kalau merge sort ini dia bakalan dibagi 2 / displit gitu. Jadi index 0 sampai 3, terus index 4 sampai 6. Nanti mereka displit lagi pokoknya. Splitnya pake rumus yang $m = \text{start} + (\text{end} - \text{start}) / 2$. Nanti kalau udah paling mentok, baru dia dibandingkan terus diurutin gitu.

Step 1: split 1 → [17, 2, 3, 8] split 2 → [5, 11, 1]

Step 2: dibagi per splitnya. [17, 2] [3, 8] [5, 11] [1]

Step 3: displit lagi [17] [2] [3] [8] [5] [11] [1]

Step 4: di merge per split yang dari step 2, yg lebih kecil di kiri. [2, 17] [3, 8] [5, 11] [1]

Step 5: merge dibandingkan sama pasangan split yg dari step 1. [2, 3, 8, 17] [1, 5, 11]

Step 6: diurutin deh dari semuanya. 1, 2, 3, 5, 8, 11, 17 //

b) Radix Sort

Di radix sort ini dia ngurutinnya kiat dari satuannya dulu, terus puluhan, dst. Nanti dikelompokin dulu yang satuannya angka 1, 2, 3, 4, 5, 6, 7, 8, 9, 0. Terus baru lanjut puluhan, dst.

kelompokin satuan →

0	1	2	3	4	5	6	7	8	9
	17	2	3		5		17	8	

kelompokin dari puluhannya →

0	1	2	3	4	5	6	7	8	9
1, 2, 3, 11,									
5, 8, 17									

1, 2, 3, 5, 8, 11, 17 //

③

index	0	1	2	3	4	5	6
value	4	6	8	10	12	17	21

ini pakai binary search yang nentuin mid, terus startnya, endnya juga, terus bandingin sama keynya.

a) 21

Step 1: mid = 3 (10) → karena 10 < 21, jadi startnya = 4, end = 6

Step 2: mid = 5 (17) → karena 17 < 21, jadi startnya = 6, end = 6

Step 3: mid = 6 (21) → karena udah sama kyk keynya (21), jadi return 6 //

b) 100

Step 1: mid = 3 (10) → 10 < 100, jadi start = 4, end = 6

Step 2: mid = 5 (17) → 17 < 100, jadi start = 6, end = 6

Step 3: mid = 6 (21) → karena ini udah index terakhir dan masih belum sama dengan keynya, startnya jadi = 7, end = 6. Dan return -1, karena gk ketemu (gaada 100 di situ).