Sorting algorithm used by computer programers

Topics
- Merge two sorted arrays
- merge sort
- Inversion Count
- Stable sorting

break –

Sorting 1 & Merge sort

P1 Given an integer array where all odd elements sorted & all even elements are sorted, sort the array.

↳ is about element not their index

Selection Sort

ex a & {2, 3, 9, 4, 15, 10, 19} → {2, 3, 4, 9, 10, 15, 19}

output

TC: O(n²)

SC: O(1)

a {2, 3

even {2, 4, 10}

odd {3, 9, 15, 19}

ans {2, 3, 4, 9, 10, 15, 19}

```
for i=0 → n-1
    for j=i → n-1
        find min(j, n-1)
    swap(a[i], a[min])
```

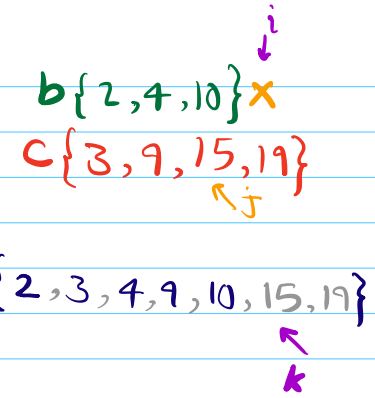
O(n+n)

TC: O(n)

SC: O(n)

P2 Merge two sorted array into one:

b[n] & c[m] → a[m+n]




```
int[] merge2(int b[], int c[]){
    n=b.Len ; i=0
    m=c.Len ; j=0
    a = new int[m+n]
    for(k=0; k<=m+n-1 ; k++){
        if( i==n)   //i    i>=n
            a[k] = c[j]
            j++
        }
        else if(j==m)  //j
            a[k] = b[i]
            i++;
        }
        else if( b[i]<=c[j] )
            a[k] = b[i]
            i++;
        }
        else{//b[i]> c[j]
            a[k] = c[j]
            j++
        }
    }
    ret a;
}
```

TC: O(n+m)

```
merge (a,start, mid, end){
SC: O(n){ b = clone(a, start, mid)
          c = clone(a, mid+1, end)
          ans = merge2(b,c)
          apply ans in a → Copy ans into a start to end
}
```

deep Copy

{9} → Sort → {9} {4} → Sort → {4} {4,9} {10,19}

how to divide?

Merge Sort divide & Conquer

ex a[] = { 9 8 7 3 6 4 1 5 0 10 } (indices 0 1 2 3 4 5 6 7 8 9)

divide

9 8 7 3 6 | 4 1 5 0 10

9 8 | 7 3 6 | 4 1 | 5 | 0 10

9 | 8 | 7 | 3 6 | 4 | 1 | 5 | 0 | 10

3 | 6 0 | 10

Conquer a.k.a merge

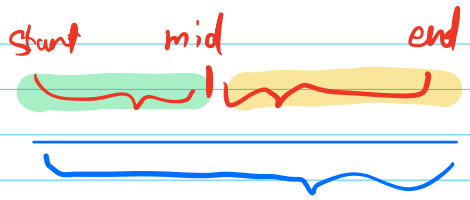
[8,9] [3,6] [1,4] [0,10]

[3,6,7] [0,5,10]

[3,6,7,8,9] [0,1,4,5,10]

[0,1,3,4,5,6,7,8,9,10]

```
void mergeSort(int a[], int start, int end){
    × if(start==end) ret
    × mid = (start+end)/2
T(n/2)  mergeSort(a, start, mid)
T(n/2)  mergeSort(a, mid+1, end)
O(n)    merge(a, start, mid, end)
}
```



$T(n) = 2T(\frac{n}{2}) + n$

$T(1) = 1$

TC : $O(n \log n)$

SC : $O(\log n + n)$

$O(n)$

will be dicussed later

P3 Given an integer array, count the number of inversion pairs in the array.

what is inversion pair ?? $i<j$ and $a[i]>a[j]$

ex a: { 8 (0), 3 (1), 4 (2) }

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

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

Quiz {4,4,4,4}

idea 1

```
for i 0→n-1
    for j←i+1 n-1
```

idea 2

Stability in Sort : relative order of equal elements should not change

$i < j, a[i] == a[j] \longrightarrow \ldots, a[i], a[j], \ldots$ (sorted →)

```
employee {
    ID,
    name,
    role
}
```

cmp based on name alphabetic order

Sort(eArr, cmp)

employee1{ ID, 123 name, "Amir" role "SDE1" }

employee2{ ID, 345 name, "Zoro" role "SDE1" }

employee3{ ID, 925 name, "Amir" role "SDE2" }

employee4{ ID, 723 name, "Bob" role "SDE1" }

sorted B

employee3{ ID, 925 name, "Amir" role "SDE2" }

employee1{ ID, 123 name, "Amir" role "SDE1" }

employee4{ ID, 723 name, "Bob" role "SDE1" }

employee2{ ID, 345 name, "Zoro" role "SDE1" }

Sort was not stable

sorted A

employee1{ ID, 123 name, "Amir" role "SDE1" }

employee3{ ID, 925 name, "Amir" role "SDE2" }

employee4{ ID, 723 name, "Bob" role "SDE1" }

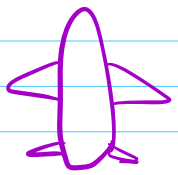
employee2{ ID, 345 name, "Zoro" role "SDE1" }

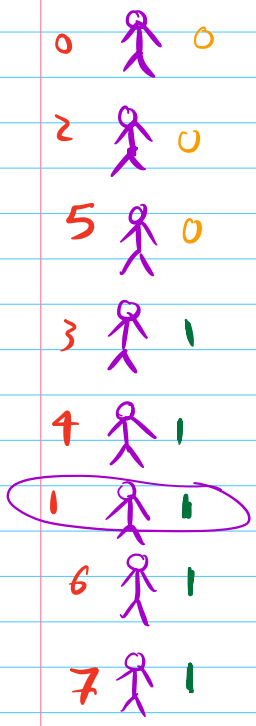
Sort was stable

Queue

0 business

1 economy





Divide & Conquer

0	1	2	3	4	5	6	7	8	9
7	8	9	3	6	4	1	5	0	10

m(a,0,9) as { 7 8 9 3 6 4 1 5 0 10 }

m(a,0,4) → 9 8 7 3 6 | 4 1 5 0 10

m(a,0,2) m(a,3,4) → 9 8 7 | 3 6 | 4 1 | 5 0 10

m(a,0,1) m(a,2,2) → 9 8 | 7 | 3 6 | 4 | 1 | 5 | 0 10

m(a,0,0) (a,1,1) → 9 8

[8,9]

[3,6]

[7,8,9]

[3,6,7,8,9]

log(n)

O(n)

TC: O(nlogn)

SC: O(logn+n)

O(n)

```
void mergeSort(int a[], int start, int end){
    if(start==end) ret
    mid = (start+end)/2
    mergeSort(a, start, mid)
    mergeSort(a, mid+1, end)
    merge(a, start, mid, end)
}
```