

1. กำหนดให้ algorithm Quick Sort แสดงดังรูปต่อไปนี้

**ALGORITHM Quicksort( $A[l..r]$ )**

//Sorts a subarray by quicksort

//Input: A subarray  $A[l..r]$  of  $A[0..n-1]$ , defined by its left

// and right indices  $l$  and  $r$

//Output: The subarray  $A[l..r]$  sorted in nondecreasing order

if  $l < r$

$s \leftarrow \text{Partition}(A[l..r])$  //  $s$  is a split position

Quicksort( $A[l..s-1]$ )

Quicksort( $A[s+1..r]$ )

**ALGORITHM Partition( $A[l..r]$ )**

//Partitions subarray by using its first element as a pivot

//Input: subarray  $A[l..r]$  of  $A[0..n-1]$ , defined by its left and right indices  $l$  and  $r$  ( $l < r$ )

//Output: A partition of  $A[l..r]$ , with the split position returned as this function's value

$p \leftarrow A[l]$

$i \leftarrow l; j \leftarrow r+1$

repeat

repeat  $i \leftarrow i+1$  until  $A[i] \geq p$

repeat  $j \leftarrow j-1$  until  $A[j] \leq p$

swap( $A[i], A[j]$ )

until  $i \geq j$

swap( $A[i], A[j]$ ) //undo last swap when  $i \geq j$

swap( $A[l], A[j]$ )

return  $j$

จาก algorithm ที่ให้มา ให้นักศึกษา แสดงผลการเรียงข้อมูลของ  $arr = [16\ 25\ 2\ 54\ 36\ 9\ 12\ 66]$  ที่ละ step มาโดยละเอียด

2. จากข้อ 1 ให้นักศึกษาเขียนโปรแกรมทั้งส่วนของ Quicksort และส่วน Partition

3. กำหนดให้ algorithm DistributionCounting แสดงดังรูปต่อไปนี้

**ALGORITHM** *DistributionCounting*( $A[0..n-1]$ )

```
//Sorts an array of integers from a limited range by distribution counting
//Input: Array  $A[0..n-1]$  of integers between  $l$  and  $u$  ( $l \leq u$ )
//Output: Array  $S[0..n-1]$  of  $A$ 's elements sorted in nondecreasing order
for  $j \leftarrow 0$  to  $u-l$  do  $D[j] \leftarrow 0$  //initialize frequencies
for  $i \leftarrow 0$  to  $n-1$  do  $D[A[i]-l] \leftarrow D[A[i]-l] + 1$  //compute frequencies
for  $j \leftarrow 1$  to  $u-l$  do  $D[j] \leftarrow D[j-1] + D[j]$  //reuse for distribution
for  $i \leftarrow n-1$  downto  $0$  do
     $j \leftarrow A[i]-l$ 
     $S[D[j]-1] \leftarrow A[i]$ 
     $D[j] \leftarrow D[j]-1$ 
return  $S$ 
```

จาก algorithm ที่ให้มา ให้นักศึกษา แสดงผลการเรียงข้อมูลของ  $arr = [16\ 25\ 2\ 54\ 36\ 9\ 12\ 66]$  ที่ละ step มาโดยละเอียด พร้อมทั้งหา  $tn$  และ  $O(n)$

4. จากข้อ 3 ให้นักศึกษาเขียนโปรแกรม
5. UnionIntersection ให้เขียนโปรแกรมสำหรับค้นหา Union และ Intersection ของสอง List ที่ต้องกรอกเข้าไป

Input:      N            =          จำนวนข้อมูลในแต่ละชุดของ List

              A[N]        =          ข้อมูลแต่ละตัวใน List 1

              B[N]        =          ข้อมูลแต่ละตัวใน List 2

Output:    บรรทัดที่ 1 = ข้อมูลที่ Intersec กันระหว่าง List 1 และ List 2

              บรรทัดที่ 2 = ข้อมูลที่ Union กันระหว่าง List 1 และ List 2

Input	Output
5 6 8 7 4 1 3 2 7 1 2	7 1 6 8 7 4 1 3 2
4 5 4 3 2 5 4 3 2	5 4 3 2 5 4 3 2
3 6 8 2 6 3 4	6 6 8 2 3 4

## 6. เรียง คั่น ยุบ [Q1SortSearchReduce]

จงเขียนโปรแกรมเรียง คั่น ยุบ ตัวอย่างเช่น อาร์เรย์  $a = \{23, 22, 18, 11, 13\}$  จากนั้นนำตัวเลขดังกล่าวมาเรียงจากน้อยไปมากจะได้  $a = \{11, 13, 18, 22, 23\}$  จากนั้นทำการต่อตัวเลขดังกล่าวให้เป็นตัวเดียว จะได้ 1113182223 จากนั้นทำ การคั่นแล้วยุบชุดของตัวเลขดังกล่าว โดยถ้าตัวเลขใดซ้ำ ให้ยุบเหลือ 1 ตัว ผลลัพธ์การเรียง คั่น ยุบ ของ  $a = \{23, 22, 18, 11, 13\}$  คือ 131823

ข้อมูลนำเข้า

บรรทัดที่ 1 แสดงจำนวนอาร์เรย์ขนาด  $n$  จำนวน

บรรทัดที่ 2 แสดงข้อมูล  $n$  จำนวนเว้นด้วยช่องว่าง

ข้อมูลส่งออก

บรรทัดที่ 1 แสดงชุดของตัวเลขจากน้อยไปมากที่นำมาต่อกัน

บรรทัดที่ 2 แสดงตัวเลขซ้ำที่ต่อเนื่องกันมากที่สุดถ้ามีมากกว่า 1 ตัวให้เรียงจากน้อยไปมาก

บรรทัดที่ 3 แสดงชุดของตัวเลขที่ถูกยุบ

ข้อมูลนำเข้า	ข้อมูลส่งออก
6 3 2 10 12 15 22 23	231012152223 2 2310121523
5 23 22 18 11 13	1113182223 1 2 131823
10 33 11 11 22 22 22 22 22 33 33	11112222222222333333 2 123
10 2 1 3 4 6 5 7 10 9 8	12345678910 0 1 2 3 4 5 6 7 8 9 12345678910
10 11 11 11 22 22 22 22 22 22 22	11111122222222222222 2 12