

Assignment 01

1.1) Input: array $A[0 \dots 49999]$ output: num of inversion

inverCount $\leftarrow 0$

```

for i  $\leftarrow 0$ ; i  $< 50000$ ; i++
  for j  $\leftarrow 0$ ; j  $< 50000$ ; j++
    if  $A[i] > A[j]$  &  $i < j$ 
      inverCount  $\leftarrow$  inverCount + 1
return inverCount

```

$\Theta(n^2)$

2.2) Input: array $A[0 \dots 49999]$ output: num of inversions

mergeSort:

inverCount $\leftarrow 0$

if ($n > 1$)

mid = $(0 + n) / 2$

copy $A[0 \dots \text{mid}]$ to $B[0 \dots \text{mid}]$

copy $A[\text{mid}+1 \dots n-1]$ to $C[\text{mid}+1 \dots n-1]$

inverCount += mergeSort($B[0 \dots \text{mid}]$)

inverCount += mergeSort($C[\text{mid}+1 \dots n-1]$)

inverCount += merge(B, C, A)

return inverCount

merge func:

```

while i ≤ mid-1 & j ≤ c
    if B[i] ≤ C[j]
        A[k] ← B[i]
        k++
    else
        A[k] ← C[j]
        invcount ← invcount + (mid - i)

if (i = mid-1)
    copy C[j ... c] to A[k ... mid-1 + c - 1]
else copy B[i ... m-1] to A[k ... mid-1 + c - 1]

```

Analysis $T(n) = 2T(n/2) + n/2$ for $n > 1$, $T(1) = 0$

$$T(n) = (n \log_2 n) / 2 \in \Theta(n \log n)$$

Master Theorem $f(n) = n/2 \in \Theta(n')$

$$a = 2, b = 2, d = 1$$

$$b^2 = 2 \quad a = 2$$

$$\therefore T(n) \in \Theta(n \log n)$$

2.2) Input: $P[0 \dots 29999]$

```
for i ← 0; i < 30000; i++  
  for j ← 0; j < 30000; j++  
    greater ← 0  
    lesser ← 0  
    if i not equal j  
      for k ← 0; k < 30000; k++  
        calculate a, b & c for  $ax + by < \text{or} > c$   
        if greater and lesser not equal to 0 do nothing  
        else  
          if k not equal to i & j  
            if  $ax + by > c$   
              greater++  
            if  $ax + by < c$   
              lesser++  
  if all points are greater or lesser print points
```

Analysis: $O(n^3)$

2.2)

If ($n < 3$)
return

else

get the most left & right most line
 quickhull(arr, size of array, left, right, 1)
 quickhull(arr, size of array, left, right, -1)

Quickhull

get the max distance & index of the point furthest from of p_1 & p_2

if no points found do
add points to the hull

quickhull(arr, size of arrays on one side, arr[index], p_1 , getside)
 quickhull(arr, size of arrays on other side, arr[index], p_2 , geside)

Analysis

$$T(n) = 2T(n/2) + n, \quad n > 1, \quad T(1) = 0$$

$$T(n) = n \log_2 n \in O(n \log n)$$

Master theorem: $f(n) = n \in \Theta(n^1)$ $a=2, b=2, d=1$

$$b^d = 2 \quad a = 2$$

$$\therefore n \log n$$