Jason Phung

Input: An array $A[0...n-1]$ output: Number of inversions

1.1 Brute Force Algorithm

for $i \leftarrow 0$ to $n-2$ do.
  for $j \leftarrow i+1$ to $n-1$ do
    if $A[i] > A[j]$ count++

$$C(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} [(n-1)-(i+1)+1]$$

$$= \sum_{i=0}^{n-2} (n-1-i)$$

$$= \frac{(n-1)n}{2} \in \Theta(n^2)$$

1.2 Recursive divide-and-conquer Algorithm

MergeMod($A[0...n-1)$ Input: An array $A[0...n-1]$, array length n
if $n > 1$ Output: Inversion count
  copy $A[0 ... \lfloor n/2 \rfloor -1]$ to $B[0 ... \lfloor n/2 \rfloor -1]$
  copy $A[\lfloor n/2 \rfloor ... n-1]$ to $C[0 ... \lceil n/2 \rceil - 1]$
  MergeMod($B[0 .. \lfloor n/2 \rfloor - 1)$
  MergeMod($C[0 ... \lceil n/2 \rceil -1)$
  Merge (B, C, A)

Input: 2 arrays $B[0...p-1]$ $C[0...q-1]$, length of both arrays: p, q
$i \leftarrow 0$; $j \leftarrow 0$; count $\leftarrow 0$
while $i < p$ and $j < q$ do
  if $B[i] \le C[j]$
    $i \leftarrow i+1$

  else
    $j \leftarrow j+1$
    count $\leftarrow$ count + $(p-i)$

1047734

1.2 $C(n) = 2C(\frac{n}{2}) + n - 1$ $C(1) = 0$

$a = 2$ $b = 2$ $f(n) = n-1 \in \Theta(n^1)$ $d = 1$

$2 = 2^1$, $\therefore$ by master theorem $T(n) \in (n \log n)$

2.1 Input: Array of points $P[0...n-1]$

Output: Number of convex and set of points

```
for i ← 0 to n-1 do
    for j ← i+1 to n do
    Calculate ax + by = C
        for k ← 0 to n-2 do
            if a(x[k]) + b(y[k]) < C
                lcount=1
            else
                rCount = 1
    If (lcount=1 and rCount=0) or (lcount=0 and rCount = 1)
    Add points to convex set
    count ← count + 1
```

$$C(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=0}^{n-2} 1 = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} [(n-2)+1] = [n-1]$$

$$= [n-1] \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1$$

$$= [n-1] \left(\frac{n(n-1)}{2}\right) \in \Theta(n^3)$$

2.2 Input: Array of points $P[\,]$, 2 different points, side

```
QuickHull(P[n], p1, p2)
for i ← 0 to n-1 do
Calculate distance from line to point p1p2
Calculate which side the point is on the line
(using ax + by = c)
If (side found == side and the distance in the greatest)
    pMax ← i   // store the index of i
If no points exist beyond the line
    Add points to the set of convex points
QuickHull(P[n], p1, P[pMax])
QuickHull(P[n], P[pMax], p2)
```

Jason Phung

2.2 $C(n) = 2C(\frac{n}{2}) + n$

$a = 2 \quad b = 2 \quad f(n) = n \in \Theta(n^1) \quad d = 1$

$2 = 2^1$, $\therefore$ by Master theorem, $C(n) \in \Theta(n \log n)$