Grace Seiche

Professor Mello Stark

CS 2223/B Term 2017

21 November 2017

Project 2

Executive Summary Report

Test Cases and Data:

Sample Set	Shortest Distance	Brute Force Time	Recursive Time
[(0,0),(7,6),(2,20),(12,5),(16,16),(5,8),(19,7),(14,22),(8,19),(7,29),(10,11),(1,13)]	2.82842712	0.000269826947	0.000218468963
[(19,18),(16,11),(15,5),(14,20),(10,6),(17,2),(7,3),(4,9),(1,8),(13,12)]	3.16227766	0.000198715892	0.000187259111
[(8,7),(8,5),(4,2),(13,2),(2,11),(18,18),(9,6)]	1.41421356	0.000119703609	0.000132345574
[(2,4),(5,8),(16,14)]	5.0	4.22715715e-05	4.069132587e-05

Time Efficiency Formula for Brute Force:

```
def brute_force(coords):
    minDist = distance(coords[0], coords[1])
    startCoord = coords[0]
    endCoord = coords[1]
    for i in range(0, len(coords)):                  // ∑_{i=0}^{n-1} 1
        for j in range (i+1, len(coords)):           //∑_{j=i+1}^{n-1} 1
            dist = distance(coords[i], coords[j])
            if(dist < minDist):                      //basic operation
                startCoord = coords[i]
                endCoord = coords[j]
                minDist = dist
    return minDist
```

$$T(n) = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-1} n - i - 1 = \frac{n(n-1)}{2}$$

$$T(n) = O(n^2)$$

The time efficiency for brute force is $O(n^2)$

Time Efficiency Formula for Recursive:

```
def EFC(p, q):
    if(len(p) <= 3):
        return brute_force(p)
    else:
        p1 = p[:len(p)//2]
        pr = p[len(p)//2:]
        q1 = q[:len(q)//2]
        qr = q[len(q)//2:]
        d1 = EFC(p1, q1)
        d2 = EFC(pr, qr)
        d = min(d1, d2)
        m = p[len(p)//2 -1].x
        s = []
        for coord in q:
            if(abs(coord.x - m) < d):
                s.append(coord)
        dminsq = pow(d, 2)
        for i in range(0, len(s)-1):
            k = i + 1
             while k <= len(s)-1 and pow(s[k].y - s[i].y, 2) < dminsq:
                dminsq = min((pow(s[k].x - s[i].x, 2) + pow(s[k].y - s[i].y, 2)), dminsq)
                k = k+1
    return math.sqrt(dminsq)
```

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

$T(n) = 2T\left(\frac{n}{2}\right) + O(3n)$

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

Master Method:

$n^{\log_2 2}\ vs\ n$

$n\ vs\ n$

Case 2: $T(n) = O(nlogn)$

The time efficiency for recursion is O(nlogn)