CSE 310 Hw1 Joshua Godinez

2.)

COST # of times

C1  n

C2 n-1

C3  (arithmetic series)

C4 Inside second for loop

C5

C6

C7

C8

C9

T(n)=O(n2) Worst Case

T(n)=O(n) Best Case (inner for loop will not run)

3.)

a. ∈ Θ()

(divide both sides by n2)

(let n=20)

let c=1 and n=20

Θ()

b. Θ() and 0() ∈ θ()

(let and)

θ()

c. 6)

(Let n=1)

let n=1 and c=19

f(n)=O(n2)

(let n=1)

let n=1 and c=17

LHS:

(let n=1)

d.

Base case:

Let n=1

n=k

Induction:

Assume n=k

therefore n=k+1 must be true

LHS:

=

=

Therefore the LHS=RHS

By induction, for n>0

4.) Power(x, n){

if(n==0)

return 1

int half=Power(x, n/2)   
if(n%2==0)

return half\*half

else   
return x\*half\*half   
}

O(logn)- Recursion called once. Using n/2 at each iteration step

5.) Closest(p)

distance=0; O(1)

n<- p.length n

for i=1 to n n

for j=i+1 to n n2

if GET\_DISTANCE(p(i),p(j))<distance

distance=Get\_Distance(p(i),p(j))

int s=p(i)

int d=p(j)

O(n2)