**CSC236: Assignment 2**

Ruijie Sun, Wenjie Hao

10th, July, 2017

**1.**

**a)**

Let T(n) = S(n) + 1

= (S(n-1))2 + 2S(n-1) + 1

= (S(n-1) + 1)2

= ((S(n-2))2 + 2S(n-2) + 1)2

= ((S(n-2) + 1)2)2

=

=

=

=

=

…

=

…

= (When k=n.)

=

=

S(n) = T(n) – 1

**= – 1 (This is the closed form of S(n))**

**Prove for the closed form.**

**Defining predicate:**

P(n): “”, where n .

**Base Case:**

Let n = 0. Proof for P(0).

LHS = S(0) = 1

RHS = = 2 – 1 = 1

Then, LHS = RHS. P(0) holds.

**Inductive steps:**

Let be an arbitrary natural number, and suppose that P() holds, i.e., .

WTS: P(n+1) holds, i.e.,

S(n+1) = (S(n))2 + 2S(n) (By given recurrence relationship.)

= (By inductive hypothesis)

=

=

=

Therefore, P(n+1) holds. The closed form is proven.

Let f(n) =

Prove that S(n)

In order to show that S(n) ∈ Θ(f(n)), we have to show that as n goes to infinity asymptotically S(n) and f(n) grow the same.

= = 1

Since, the above limit is less than infinity and more than 0, we have that S(n) ∈ Θ(f(n)).

Therefore, the bound is correct.

**b)**

**Case 1: n is even and n > 1**

S(n) = S(n-2) + 2n -1

= S(n-4) + 2(n-2) -1 + 2n -1

= S(n-6) + 2(n-4) -1 + 2(n-2) -1 + 2n -1

=S(n-6) + 2((n-4) + (n-2) + n) -3

…

=S(n-k) + 2((n-k+2) + (n-k+4) + …+n) -

…

=S(n-n) + 2((n-n+2) + (n-n+4) + …+n) -

= S(0) + 2(2 + 4 + …+n) -

= 0 + 2 -

**=**  +

**Case 2: n is odd and n >1:**

S(n) = S(n-2) + 3n

= S(n-4) + 3(n-2) + 3n

= S(n-6) + 3(n-4) + 3(n-2) + 3n

…

= S(n-k) + 3[(n-k+2) + (n-k+4) + …+ n]

…

= S(n-(n-1)) + 3[(n-(n-1)+2) + (n-(n-1)+4)+ … + n]

= S(1) + 3[3+5+…+n]

= 1 + 3

= 1 + (n+3)(n-1)

= + n -

**Prove for the closed form.**

**Defining predicate: P(n):** “, where n .

Base Case 1: n = 0.

LHS = S(0) = 0

RHS = = 0

Since LHS = RHS, P(0) holds.

Base Case 2: n = 1.

LHS = S(1) = 1

RHS = = 1

Since LHS = RHS, P(1) holds.

**Inductive steps:**

Let n , n > 1. Assume H(n) : ∀ i , 0 ≤ i ≤ n, P(i)

We want to show P(n).

**Case 1: n is even.**

Since n > 1, then n – 2 ≥ 0.

Then 0 ≤ n-2 < n, by Induction hypothesis, P(n-2) holds.

Since (n-2) is even, we have S(n-2) = + .

S(n) = S(n-2) + 2n -1 (By given recursive relationship)

= + + 2n – 1 (By inductive hypothesis)

= +

=

Then P(n) holds.

**Case 2: n is odd**

Since n > 1, then n – 2 ≥ 0.

Then 0 ≤ n-2 < n, by Induction hypothesis, P(n-2) holds.

Since (n-2) is odd, S(n-2) = .

S(n) = S(n-2) + 3n (By given recursive relationship)

= + 3n

= - 3 + 3n

=

Then P(n) holds.

Therefore, the closed form is correct.

Let f(n) =

Prove that S(n)

In order to show that S(n) ∈ Θ(f(n)), we have to show that as n goes to infinity asymptotically S(n) and f(n) grow the same.

Case 1: n is even

= =

Since, the above limit is less than infinity and more than 0, we have that S(n) ∈ Θ(f(n)).

Therefore, the bound is correct for all even natural number.

Case 2 : n is odd

= =

Since, the above limit is less than infinity and more than 0, we have that S(n) ∈ Θ(f(n)).

Therefore, the bound is correct for all odd natural number.

Therefore, the bound is correct for all natural number.

**2.**

Prove for special case: n = , for some k

S(n) = + 3n – 5 (for this particular special case)

Let f(n) =

In order to show that S(n) ∈ Θ(f(n)) for this special case, we have to show that as n goes to infinity asymptotically S(n) and f(n) grow the same.

= = 1

Since, the above limit is less than infinity and more than 0, we have that S(n) ∈ Θ(f(n)) for this special case. Therefore, the bound is correct for this special case when n = , for some k .

**Proving for general:**

Let n , n>1.

Case 1:  k , n =

Then S(n) ∈ Θ() (we have proved it above)

Case 2: ∀ k , n ≠

**By Fact 1,  k , ≤ n ≤.**

**Then ≤ 2n, and ≤ .**

Since S(n) is monotonic non-decreasing,

then ≤S(n) ≤.

In order to prove S(n)(), we need to prove S(n) ∈ **and** S(n) ∈

Prove S(n) ∈: i.e,  c1 +,  n0 , ∀n nn0 S(n) ≤ c1

Let n0 = 64, c1 = 4

Let n nn0

S(n) ≤)

= 2klog(2k) + 32k – 5 (by given information)

log(2n) + 32n)– 5 (since ≤ 2n)

log(2n) + 6n

= 2n

= 2n + n(2log(2) + 6)

= 2n+ n(log4 + 6log(2)) (since log2 = 1)

2n(because )

2n (because )

= 4

= c1 (since c1 = 4)

Then, S(n) ∈.

Prove S(n) ∈:i.e,  c2 +,  n1 , ∀n nn1 S(n) c2

Let n1 = 20, c2 =

Let n nn1

S(n) = 2k-1log(2k-1) + 32k-1 – 5 (by given information)

log() + 3– 5log(n) (since )

log(n) - log(2)) + 3– 5log(n)

nlog(n) - log(2) + 3– 10log(n))

= nlog(n) - log(2) + 3log(2)– 10log(n))

nlog(n) + 2log(2)– 10log(n))

nlog(n) – 10log(n)) (since 2log(2) > 0)

nlog(n) – log(n)) (since n>20, then )

n log(n)))

n log(n))

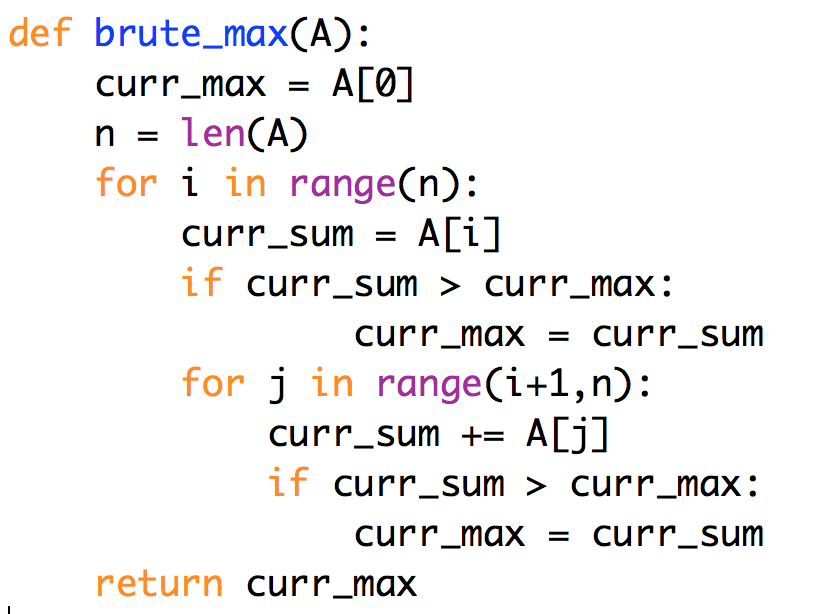
n log(n))

Then S(n) ∈

Therefore, S(n)()

**3.**

**a)**

****

Let n be the length of the input A, which is a natural number.

For a fixed iteration of the outer loop, the inner loop runs n-1-i iteration(s), with each iteration taking constant time. The outer loop runs n iterations, for i going from 0 to n – 1. The total cost is

= n(n-1) - = , which ∈**.**

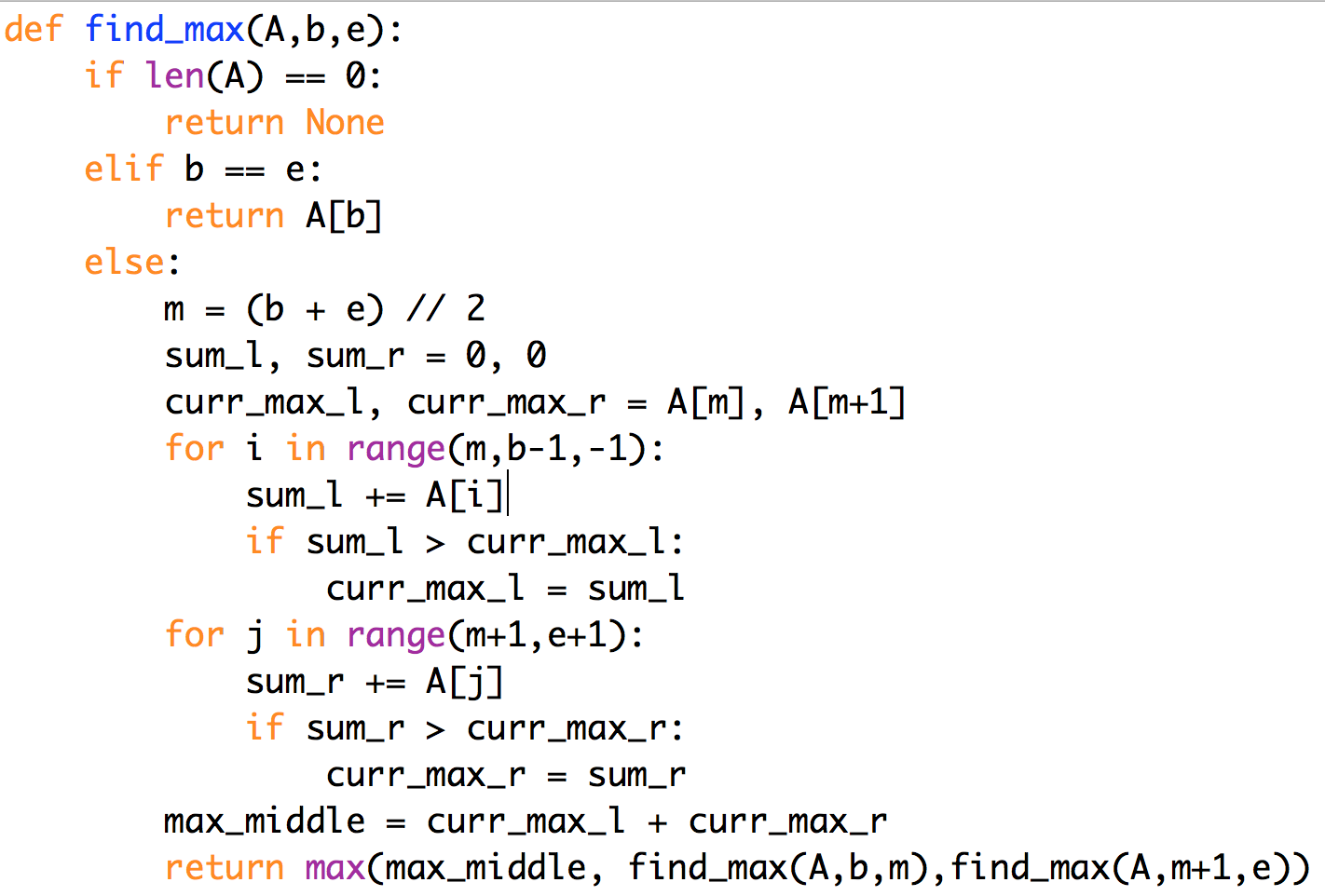
**b)**

**T**

where n.

**a = 2, b=2**

**c)**



**d)**

Using master theorem to prove that T(n)().

Cost of splitting and recombining: f(n) = n + 1

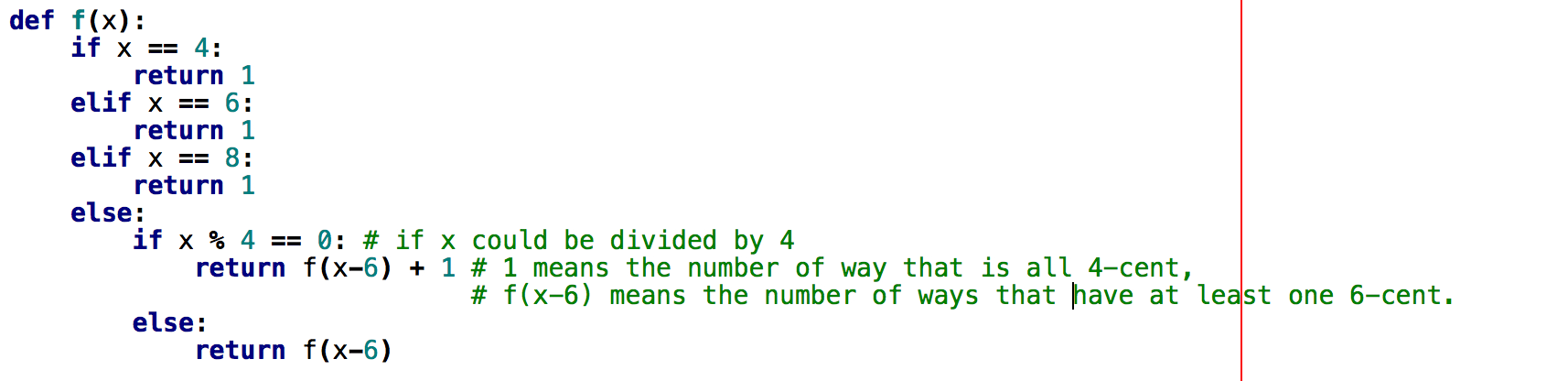
The cost of splitting and combining (d), which is () for T(n). This means, d = 1.

Since a=2, b=2, d=1, a = bd, so T(n)().

Therefore, T(n) )().

4.

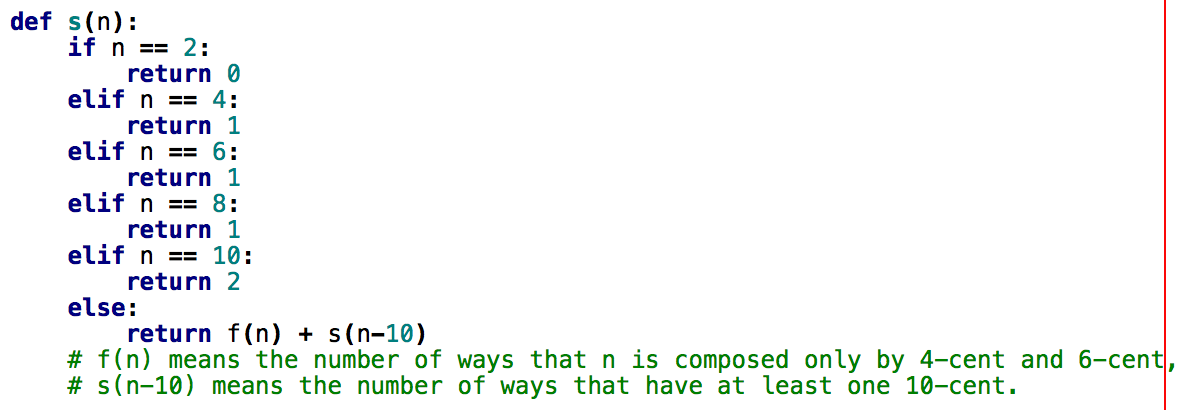
a)



Let n=4, n=6, n=8 as base cases.

For all even number which is greater than 6, we classify it by whether or not it can be divided by 4 with no remainder. If it can be divided by four with no remainder, we can divide the ways of construction into two different group with no overlap. The first group is the way that this number is composed only by 4-cent. The second group is the set that each element of the set has at least one 6-cent. In other words, the element number of group 2 is f(n-6). If it can not be divided by four with no remainder, it must have at least one 6-cent, so the number of construction is f(n-6).

b)



Let H(n) denote the number of way that a postage of n cents, where n 4 and n is even, can be only made by 4-cent stamps.

H(n) =

Let T(n) denote the number of distinct ways that a postage of n cents, where n 4 and n is even, can be made by 4-cent and 6-cent stamps.

T(n) =

Let S(n) denote the number of distinct ways that a postage of n cents, where n 4 and n is even, can be made by 4-cent, 6-cent and 10-cent stamps.

S(n) =

c)

Let Q(n) denote the number of way that a postage of n cents, where n 4 and n is even, can be only made by 6-cent stamps.

Q(n) =

Define M(n) which is equivalent to T(n):

M(n) =

Defining predicate:

P(n): “S(n) S(n-2)”, where n, n4 and n is even.

Base Case 1: n = 4

P(4) is vacuous true since there is no S(2) to compare.

Base Case 2: n = 6

S(6) = 1

S(4) = 1

S(6) = S(4) , then P(6) holds.

Base Case 3: n = 8

S(8) = 1

S(6) = 1

S(8) = S(6) , then P(8) holds.

Base Case 4: n = 10

S(10) = 2

S(8) = 1

S(10) > S(8) , then P(10) holds.

Base Case 5: n = 12

S(12) = 2

S(10) = 2

S(12) = S(10) = 2, then P(12) holds.

Base Case 6: n = 14

S(14) = (1+1) = 2

S(12) = 2

S(14) = S(12), then P(14) holds.

Base Case 6: n = 16

S(16) = (2+1) = 3

S(14) = 2

S(16) > S(14), then P(16) holds.

Base Case 6: n = 18

S(18) = 3

S(16) = 3

S(18) = S(16), then P(18) holds.

Base Case 6: n = 20

S(20) = 4

S(18) = 3

S(20) > S(18), then P(20) holds.

Base Case 6: n = 22

S(22) = 4

S(20) = 4

S(22) = S(20), then P(22) holds.

Base Case 6: n = 24

S(24) = 5

S(22) = 4

S(24) = S(22), then P(24) holds.

**Inductive Steps:**

Let n , n and n is an even number.

Assume H(n): ∀ i , 4 , i.e., S(i) S(i-2)

WTS: P(n).

S(n) – S(n-2) = (T(n) + S(n-10)) – (T(n-2) + S(n-12))

= (T(n) – T(n-2)) + (S(n-10) – S(n-12)

= (T(n) – T(n-2)) + (T(n-10) + S(n-20) – T(n-12) – S(n-22))

T(n) – T(n-2) + T(n-10) – T(n-12)

(By Inductive hypothesis, 4 both (n-22) and (n-20) are even number, then S(n-20)

= T(n) + T(n-10) – T(n-2) - T(n-12)

= H(n) + T(n-6) + H(n-10) + T(n-16) – T(n-2) - T(n-12)

(By the definition of T(n))

= H(n) + T(n-6) + H(n-10) + T(n-16) - M(n-2) – M(n-12)

(Since M(n) is equivalent to T(n))

= H(n) + T(n-6) + H(n-10) + T(n-16) - Q(n-2) – M(n-6) – Q(n-12) – M(n-16)

= H(n) + H(n-10) - Q(n-2) – Q(n-12) + (T(n-6) –M(n-6)) + (T(n-16) – M(n-16))

= H(n) + H(n-10) - Q(n-2) – Q(n-12)

(Since T(n) is equivalent to M(n))

= H(n) + H(n-10) - (Q(n-2) + Q(n-12))

**Case 1: (n mod 4) = 2 and (n mod 6) = 0**

H(n) + H(n-10) - (Q(n-2) + Q(n-12)) = 0 + 1 – (0 + 1) = 0

Then, case 1 holds.

**Case 2: (n mod 4) = 2 and (n mod 6) 0**

H(n) + H(n-10) - (Q(n-2) + Q(n-12)) = 0 + 1 – (Q(n-2) + 0) = 1 – Q(n-2)

(Since Q(n-2) = 1 or 0)

Then, case 2 holds.

**Case 3: (n mod 4) 0 and (n mod 6) 0**

H(n) + H(n-10) - (Q(n-2) + Q(n-12)) = 1 + 0 – (0 + 1) = 0

Then, case 3 holds.

**Case 4: (n mod 4) 0 and (n mod 6) 0**

H(n) + H(n-10) - (Q(n-2) + Q(n-12)) = 1 + 0 – (Q(n-2) + 0) = 1 – Q(n-2)

(Since Q(n-2) = 1 or 0)

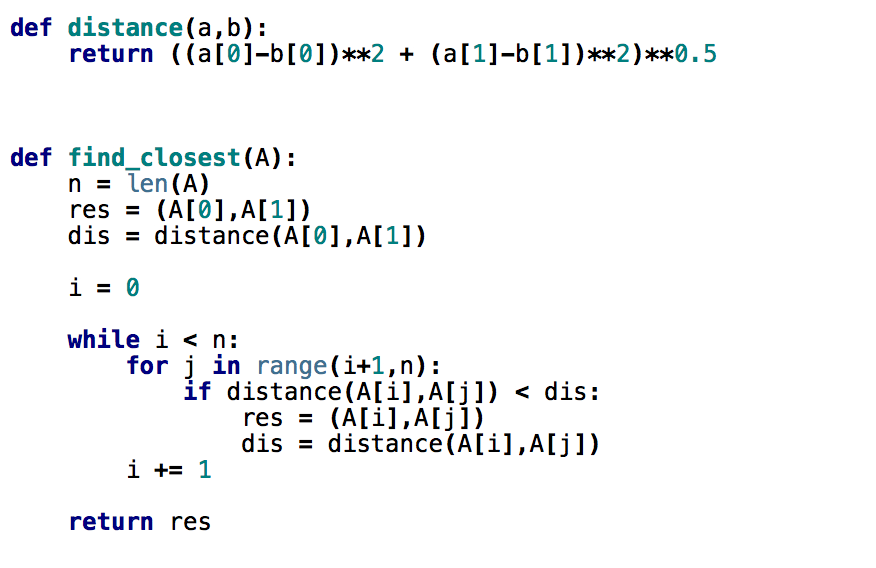
Then, case 4 holds.

All cases hold, so P(n) holds.

Therefore, S(n) is non-decreasing.

5.

a)



b)

Let n be the length of the input A, which is a natural number.

For a fixed iteration of the outer loop, the inner loop runs n-1-i iteration(s), with each iteration taking constant time. The outer loop runs n iterations, for i going from 0 to n – 1. The total cost is

= n(n-1) - = , which ∈.

c)

Let n be the length of input A and n.

**Outer Loop Invariant:**

* i n,
* ∀ m 0mi-1, ∀s , 0sn-1, m s res distance(A[m], A[s])

**Inner Loop Invariant:**

* i j n
* ∀, ipj-1, res distance(A[i-1], A[p])

**Prove for inner loop’s partial correctness.**

For a fixed i-th outer loop, assume the outer loop invariant holds for (i-1)-th iteration.

**Base Case:**

At the end of 0-th iteration of inner loop,

j0 = i, then i j0 n

Since j0 – 1 = i -1 < i, then ∀, ipj-1, res distance(A[i-1], A[p]) is vacuous true.

**Inductive Steps:**

Let jk , where k

At the end of k-th iteration of inner loop,

jk n and ∀, ip jk - 1, res distance(A[i-1], A[p]) is true.

We want to show that at the end of (k+1)-th iteration of inner loop, ∀, ip jk+1 - 1, res distance(A[i-1], A[p]) is true.

**Case 1:** there is no (k+1)-th iteration of inner loop, then jk+1 = jk, then jk+1 n and ∀, ip jk+1 - 1, res distance(A[i-1], A[p]) is true.

**Case 2:** there is (k+1)-th iteration of inner loop, then jk n, then jk+1 = jk+1, then

jk+1 n.

***Subcase 1:*** res updates because the condition of the if-statement in the inner loop is triggered, i.e., distance(A[i-1], A[jk+1-1]) < dis #dis is the previous minimum distance of two points which we have passed.

***Subcase 2:*** res doesn’t update because distance(A[i-1], A[jk+1-1]) dis.

By inductive hypothesis, ∀, ip jk - 1, res distance(A[i-1], A[p]), combining with the result from subcase 1 and subcase 2,

jk+1 n and ∀, ip jk+1 - 1, res distance(A[i-1], A[p]).

**Prove for termination of inner loop:**

Let Ek = n – jk. By loop invariant, jk. So Ek Ek, ∀k.

If there is a k+1 iteration,then Ek+1 = n - jk+1 = n - (jk + 1) < n - jk = Ek.

Thus, the inner loop terminates.

**Prove for outer loop’s partial correctness.**

Base Case:

Since, i0 = 0

Then i0 n,

Since, i0 – 1 < 0, then,

∀ m 0mi-1, ∀s , 0sn-1, m s res distance(A[m], A[s]) is vacuous true.

**Inductive Steps:**

Let ik , where k .

At the end of k-th iteration of outer loop,

assume that ik n and

∀ m 0m ik - 1, ∀s , 0sn-1, m s res distance(A[m], A[s])

WTS: at the end of (k+1)th iteration of outer loop, ik+1 n and ∀ m 0m ik+1 - 1, ∀s , 0sn-1, m s res distance(A[m], A[s])

**Case 1:** there is no (k+1)th iteration of outer loop, then ik+1 =ik  n and

∀ m 0m ik+1 - 1, ∀s , 0sn-1, m s res distance(A[m], A[s])

(By Induction Hypothesis)

**Case 2:** there is (k+1)th iteration of outer loop, it means ik n, then ik+1 n.

Assume that at the end of (k+1)-th iteration of outer loop, the post-condition of inner loop is correct, i.e., ∀, ik+1p n-1, res distance(A[ik+1-1], A[p]).

Combining with Induction Hypothesis, we can get:

∀ m 0m ik+1 - 1, ∀s , 0sn-1, m s res distance(A[m], A[s])

**Prove for outer loop’s termination.**

Let Ek = n – ik. By loop invariant, ik. So Ek Ek, ∀k.

If there is a k+1 iteration,then Ek+1 = n - ik+1 = n - (ik + 1) < n - jk = Ek.

Thus, the outer loop terminates.

Since, the partial correctness of inner loop and outer loop are both correct, and the inner loop and outer loop both terminate, the algorithm is correct.