CSC236: Assignment 2

Ruijie Sun, Wenjie Hao

10^{th}, July, 2017

1.

a)

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

$= (S(n-1))^2 + 2\times S(n-1) + 1$

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

$= ((S(n-2))^2 + 2\times S(n-2) + 1)^2$

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

$= (S(n-2) + 1)^2 \times (S(n-2) + 1)^2$

$=(S(n-2) + 1)^{2^2}$

$= (S(n-3)^2 + 2\times S(n-3) \ + 1)^{2^2}$

$= (S(n-3) + 1)^2)^{2^2}$

$=(S(n-3) + 1)^{2^3}$

...

$= (S(n-k) + 1)^{2^k}$

…

$= (S(n-n)+1)^{2^n}$ (When k=n.)

$= (S(0)+1)^{2^n}$

$= 2^{2^n}$

$S(n) = T(n) - 1$

$= 2^{2^n} - 1$ **(This is the closed form of S(n))**

<u>Prove for the closed form.</u>

Defining predicate:

P(n): "$S(n) = 2^{2^n} - 1$", where $n \in \mathbb{N}$.

Base Case:

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

LHS = S(0) = 1

RHS = $2^{2^0} - 1$ = 2 – 1 = 1

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

Inductive steps:

Let n be an arbitrary natural number, and suppose that P(n) holds, i.e., $\mathrm{S(n)} = 2^{2^n} - 1$.

WTS: P(n+1) holds, i.e., $\mathrm{S(n+1)} = 2^{2^{n+1}} - 1$

$S(n+1) = (S(n))^2 + 2\times S(n)$ (By given recursive relationship.)

$= (2^{2^n} - 1)^2 + 2\times(2^{2^n} - 1)$ (By inductive hypothesis)

$= (2^{2^n})^2 - 2\times 2^{2^n} + 1 + 2\times 2^{2^n} - 2$

$= (2^{2^n})^2 - 1$

$= 2^{2^{n+1}} - 1$

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

Let $f(n) = 2^{2^n}$

Prove that $S(n) \in \Theta(f(n))$

In order to show that $S(n) \in \Theta(f(n))$, we have to show that as n goes to infinity asymptotically S(n) and f(n) grow the same.

$$\lim_{n\to\infty}\left(\frac{S(n)}{f(n)}\right) = \lim_{n\to\infty}\frac{2^{2^n}-1}{2^{2^n}} = 1$$

Since, the above limit is less than infinity and more than 0, we have that $S(n) \in \Theta(f(n))$.

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) - $\frac{k}{2}$

…

=S(n-n) + 2((n-n+2) + (n-n+4) + …+n) - $\frac{n}{2}$

= S(0) + 2(2 + 4 + …+n) - $\frac{n}{2}$

= 0 + 2$\times\frac{(2+n)\times\frac{n}{2}}{2}$ - $\frac{n}{2}$

$= \frac{n^2}{2} + \frac{n}{2}$

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]

$= S(1) + 3[3+5+\ldots+n]$

$= 1 + 3 \times \frac{(3+n)(\frac{n-1}{2})}{2}$

$= 1 + \frac{3}{4}(n+3)(n-1)$

$= \frac{3}{4}n^2 + \frac{3}{2}n - \frac{5}{4}$

<u>Prove for the closed form.</u>

Defining predicate: P(n): " $\boldsymbol{S(n)} = \begin{cases} \frac{n^2}{2} + \frac{n}{2} & \text{if n is even.} \\ \frac{3}{4}n^2 + \frac{3}{2}n - \frac{5}{4} & \text{if n is odd.} \end{cases}$ ", where $n \in \mathbb{N}$.

Base Case 1: n = 0.

LHS = S(0) = 0

RHS = $\frac{0^2}{2} + \frac{0}{2} = 0$

Since LHS = RHS, P(0) holds.

Base Case 2: n = 1.

LHS = S(1) = 1

RHS = $\frac{3}{4} + \frac{3}{2} - \frac{5}{4} = 1$

Since LHS = RHS, P(1) holds.

Inductive steps:

Let $n \in \mathbb{N}$, $n > 1$. Assume $H(n) : \forall i \in \mathbb{N}, 0 \leq i \leq n, P(i)$

We want to show P(n).

Case 1: n is even.

Since $n > 1$, then $n - 2 \geq 0$.

Then $0 \leq n-2 < n$, by Induction hypothesis, P(n-2) holds.

Since (n-2) is even, we have $S(n-2) = \frac{(n-2)^2}{2} + \frac{n-2}{2}$.

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

$= \frac{(n-2)^2}{2} + \frac{n-2}{2} + 2n - 1$ (By inductive hypothesis)

$= \frac{n^2 - 4n + 4}{2} + \frac{5n-4}{2}$

$= \frac{n^2 + n}{2}$

Then P(n) holds.

Case 2: n is odd

Since $n > 1$, then $n - 2 \geq 0$.

Then $0 \leq n-2 < n$, by Induction hypothesis, P(n-2) holds.

Since (n-2) is odd, $S(n-2) = \frac{3}{4}(n-2)^2 + \frac{3}{2}(n-2) - \frac{5}{4}$.

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

$= \frac{3}{4}(n-2)^2 + \frac{3}{2}(n-2) - \frac{5}{4} + 3n$

$$= \frac{3}{4}(n^2 - 4n + 4) + \frac{3}{2} n - 3 - \frac{5}{4} + 3n$$

$$= \frac{3}{4}n^2 + \frac{3}{2} n - \frac{5}{4}$$

Then P(n) holds.

Therefore, the closed form is correct.

Let $f(n) = n^2$

Prove that $S(n) \in \Theta(f(n))$

In order to show that $S(n) \in \Theta(f(n))$, we have to show that as n goes to infinity asymptotically S(n) and f(n) grow the same.

Case 1: n is even

$$\lim_{n\to\infty}\left(\frac{S(n)}{f(n)}\right) = \lim_{n\to\infty}\frac{\frac{n^2+n}{2}}{n^2} = \frac{1}{2}$$

Since, the above limit is less than infinity and more than 0, we have that $S(n) \in \Theta(f(n))$.

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

Case 2 : n is odd

$$\lim_{n\to\infty}\left(\frac{S(n)}{f(n)}\right) = \lim_{n\to\infty}\frac{\frac{3}{4}n^2 + \frac{3}{2}n - \frac{5}{4}}{n^2} = \frac{3}{4}$$

Since, the above limit is less than infinity and more than 0, we have that $S(n) \in \Theta(f(n))$.

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 = 2^k, for some k ∈ ℕ.

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

Let f(n) = n $\log n$

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.

$$\lim_{n\to\infty}\left(\frac{S(n)}{f(n)}\right) = \lim_{n\to\infty}\frac{\text{n}\log n + 3\text{n} - 5}{\text{n}\log n} = 1$$

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 = 2^k, for some k ∈ ℕ.

Proving for general:

Let n ∈ ℕ, n>1.

Case 1: ∃ k ∈ ℕ, n = 2^k

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

Case 2: ∀ k ∈ ℕ, n ≠ 2^k

By Fact 1, ∃ k ∈ ℕ, $2^{k-1} \leq n \leq 2^k$.

Then $2^k \leq 2n$, and $\frac{n}{2} \leq 2^{k-1}$.

Since S(n) is monotonic non-decreasing,

then $S(2^{k-1}) \leq S(n) \leq S(2^k)$.

In order to prove $S(n) \in \Theta(n \log n)$, we need to prove $S(n) \in \mathbf{O}(n \log n)$ and $S(n) \in \mathbf{\Omega}(n \log n)$

Prove $S(n) \in O(n \log n)$: i. e, $\exists\, c_1 \in R^+, \exists\, n_0 \in \mathbb{N}, \forall n \in \mathbb{N},\ n \geq n_0 \Longrightarrow S(n) \leq c_1 (n \log n)$.

Let $n_0 = 64$, $c_1 = 4$

Let $n \in \mathbb{N},\ n \geq n_0$

$S(n) \leq S(2^k)$

$= 2^k \times \log 2^k + 3\times 2^k - 5$ (by given information)

$\leq 2n \times \log(2n) + 3\times(2n) - 5$ (since $2^k \leq 2n$)

$\leq 2n \times \log(2n) + 6n$

$= 2n \times \log(2) + 2n\log(n) + 6n$

$= 2n \times \log(n) + n(2\times\log(2) + 6)$

$= 2n \times \log(n) + n(\log 4 + 6 \times \log(2))$ (since $\log 2 = 1$)

$\leq 2n \times \log(n) + n(\log n + \log(2^6))$ (because $n \geq 64, so\ n \geq 4$)

$\leq 2n \times \log(n) + n(\log n + \log(n))$ (because $n \geq 64$)

$= 4 \times (n \log n)$

$= c_1 (n \log n)$ (since $c_1 = 4$

Then, $S(n) \in O(n \log n)$.

Prove $S(n) \in \mathbf{\Omega}(n \log n)$: i. e, $\exists\, c_2 \in R^+, \exists\, n_1 \in \mathbb{N}, \forall n \in \mathbb{N},\ n \geq n_1 \Longrightarrow S(n) \geq c_2 (n \log n)$.

Let $n_1 = 20$, $c_2 = \frac{1}{2}$

Let $n \in \mathbb{N},\ n \geq n_1$

$S(n) \geq S(2^{k-1}) = 2^{k-1} \times \log(2^{k-1}) + 3\times 2^{k-1} - 5$ (by given information)

$\geq \frac{n}{2} \times \log(\frac{n}{2}) + 3\times\frac{n}{2} - 5\times\log(n)$ (since $\frac{n}{2} \leq 2^{k-1}$)

$= (\frac{n}{2} \times \log(n) - \frac{n}{2} \times \log(2)) + 3\times\frac{n}{2} - 5\times\log(n)$

$= \frac{1}{2}(n\times\log(n) - n\times \log(2) + 3n - 10\times\log(n))$

$= \frac{1}{2}(n\times\log(n) - n\times \log(2) + 3n\times \log(2) - 10\times\log(n))$

$= \frac{1}{2}(n\times\log(n) + 2n\times \log(2) - 10\times\log(n))$

$\geq \frac{1}{2}(n\times\log(n) - 10\times\log(n))$ (since $2n\times \log(2) > 0$)

$\geq \frac{1}{2}(n \log(n) - \frac{1}{2}\times n \log(n))$ (since n>20, then $\frac{1}{2}n > 10$)

$\geq \frac{1}{2}(\frac{1}{2}\times(n \log(n)))$

$\geq \frac{1}{4}(n \log(n))$

$= c_2(n \log(n))$

Then $S(n) \in \boldsymbol{\Omega}(n \log n)$

Therefore, $S(n) \in \Theta(n \log n)$

3.

a)

```
def brute_max(A):
    curr_max = A[0]
    n = len(A)
    for i in range(n):
        curr_sum = A[i]
        if curr_sum > curr_max:
                curr_max = curr_sum
        for j in range(i+1,n):
            curr_sum += A[j]
            if curr_sum > curr_max:
                curr_max = curr_sum
    return curr_max
```

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

$\sum_{i=0}^{n-1}(n-1-i)$ = n(n-1) - $\sum_{i=0}^{n-1} i = \frac{\text{n(n-1)}}{2}$, which $\in \mathbf{O}(\text{n}^2)$.

b)

$$\mathbf{T(n)} = \begin{cases} \mathbf{1} & \boldsymbol{if\ n = 0} \\ \mathbf{1} & \boldsymbol{if\ n = 1} \\ \boldsymbol{T\left(\left\lceil \frac{n}{2} \right\rceil\right) + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n + 1} & \boldsymbol{if\ n > 1} \end{cases}$$

where n∈ ℕ.

a = 2, b=2

c)

```
def find_max(A,b,e):
    if len(A) == 0:
        return None
    elif b == e:
        return A[b]
    else:
        m = (b + e) // 2
        sum_l, sum_r = 0, 0
        curr_max_l, curr_max_r = A[m], A[m+1]
        for i in range(m,b-1,-1):
            sum_l += A[i]
            if sum_l > curr_max_l:
                curr_max_l = sum_l
        for j in range(m+1,e+1):
            sum_r += A[j]
            if sum_r > curr_max_r:
                curr_max_r = sum_r
        max_middle = curr_max_l + curr_max_r
        return max(max_middle, find_max(A,b,m),find_max(A,m+1,e))
```

d)

Using master theorem to prove that T(n) ∈ Θ(n log n).

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

The cost of splitting and combining ∈ $\Theta(n^{d})$, which is ∈ $\Theta(n)$ for T(n). This means, d = 1.

Since a=2, b=2, d=1, a = b^{d}, so T(n) ∈ $\Theta(n^{d}\log n)$.

Therefore, T(n)) ∈ $\Theta(\text{n}\log n)$.

4.

a)

```
def f(x):
    if x == 4:
        return 1
    elif x == 6:
        return 1
    elif x == 8:
        return 1
    else:
        if x % 4 == 0: # if x could be divided by 4
            return f(x-6) + 1 # 1 means the number of way that is all 4-cent,
                              # f(x-6) means the number of ways that have at least one 6-cent.
        else:
            return f(x-6)
```

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)

```python
def s(n):
    if n == 2:
        return 0
    elif n == 4:
        return 1
    elif n == 6:
        return 1
    elif n == 8:
        return 1
    elif n == 10:
        return 2
    else:
        return f(n) + s(n-10)
    # f(n) means the number of ways that n is composed only by 4-cent and 6-cent,
    # s(n-10) means the number of ways that have at least one 10-cent.
```

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.

$$T(n) = \begin{cases} 1 & if\ n = 4, 6, 8 \\ T(n-6) + 1 & if\ n > 10\ and\ (n\ mod\ 4) = 0 \\ T(n-6) & if\ n > 10\ and\ (n\ mod\ 4)\ \neq 0 \end{cases}$$

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.

$$S(n) = \begin{cases} 1 & if\ n = 4, or\ n = 6, or\ n = 8 \\ 2 & if\ n = 10 \\ 2 & if\ n = 12 \\ T(n) + S(n-10) & if\ n > 12 \end{cases}$$

c)

By unwinding T(n), we can get closed form of T(n) as below:

$$T(n) = \begin{cases} \left\lfloor \frac{n}{12} \right\rfloor & if\ (n\ mod\ 12) = 2 \\ \left\lfloor \frac{n}{12} \right\rfloor + 1 & if\ (n\ mod\ 12) \neq 2 \end{cases}$$

Using complete induction to prove the non-decreasing property of S(n).

Defining predicate:

P(n): "S(n) ≥ S(n-2)", where n∈ ℕ, n≥4 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.

$S(6) = S(4)$, 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.

Inductive Steps:

Let n ∈ ℕ, n > 14 and n is an even number.

Assume H(n): $\forall$ i ∈ ℕ, 4 ≤ $i < n$ *and i is even number , then* $P(i)$, i.e., S(i) ≥ S(i-2)

WTS: P(n).

Case 1: (n mod 12) ≠2:

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)

(Since n>14 and n is a even number, $4 \leq n\text{-}12 < n\text{-}10 < n$, and (n-12) and (n-10) are both even number, then S(n-10) ≥ S(n-12) by inductive hypothesis.)

≥ T(n) – T(n-2)

= $\left(\left\lfloor\frac{n}{12}\right\rfloor + 1\right)$ - T(n-2)

subcase 1: (n-2) % 12 $= 0$ *or* (n-2) % 12 = 4

$$\left(\left\lfloor\frac{n}{12}\right\rfloor + 1\right) - T(n-2) = \left(\left\lfloor\frac{n}{12}\right\rfloor + 1\right) - \left(\left\lfloor\frac{n-2}{12}\right\rfloor + 1\right) = 0$$

subcase 2: (n-2) % 12 $\neq 0$ *and* (n-2) % 12 ≠ 4:

$$\left(\left\lfloor\frac{n}{12}\right\rfloor + 1\right) - T(n-2) = \left(\left\lfloor\frac{n}{12}\right\rfloor + 1\right) - \left\lfloor\frac{n-2}{12}\right\rfloor = 1$$

Case 2: (n mod 12) =2

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)

= $\left\lfloor\frac{n}{12}\right\rfloor - \left\lfloor\frac{n-2}{12}\right\rfloor - 1$ + S(n-10) – S(n-12)

= -1 + S(n-10) – S(n-12)

Define P'(n): "S(n-10) – S(n-12) ≥ 1" ,where n ∈ ℕ, $n > 14$, $(n \bmod 12) = 2$

Base Case:

n = 26, S(26-10) =S(16) = 2

S(26-12) = S(14) = 1

Then S(16) – S(14) ≥ 1

Induction Steps:

Let n ∈ ℕ, $n > 14$, $(n \bmod 12)$ = 2. Assume P'(n), i.e.,S(n-10) – S(n-12) ≥ 1

WTS:P'(n+12), i.e., S(n+2) – S(n) ≥ 1

We don't know how to do remaining induction steps.

After this part, we can know P(n) holds.

Therefore, S(n) is non-decreasing.

5.

a)

```
def distance(a,b):
    return ((a[0]-b[0])**2 + (a[1]-b[1])**2)**0.5


def find_closest(A):
    n = len(A)
    res = (A[0],A[1])
    dis = distance(A[0],A[1])

    i = 0

    while i < n:
        for j in range(i+1,n):
            if distance(A[i],A[j]) < dis:
                res = (A[i],A[j])
                dis = distance(A[i],A[j])
        i += 1

    return res
```

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

$\sum_{i=0}^{n-1}(n-1-i)$ = n(n-1) - $\sum_{i=0}^{n-1} i = \frac{n(n-1)}{2}$, which $\in O(n^2)$.

c)

Let n be the length of input A and n$\in \mathbb{N}$.

Outer Loop Invariant:

- $i \leq n$,
- $\forall$ m $\in \mathbb{N}$, $0 \leq m \leq i-1$, $\forall$ s $\in \mathbb{N}$, $0 \leq s \leq n-1$, m $\neq$ s $\Longrightarrow$ res $\leq$ distance(A[m], A[s])

Inner Loop Invariant:

- $i \leq j \leq n$
- $\forall p \in \mathbb{N}$, $i \leq p \leq j-1$, res $\leq$ 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,

$j_0 = i$, then $i \leq j_0 \leq n$

Since $j_0 - 1 = i - 1 < i$, then $\forall p \in \mathbb{N}$, $i \leq p \leq j-1$, res $\leq$ distance(A[i-1], A[p]) is vacuous true.

Inductive Steps:

Let $j_k \in \mathbb{N}$, where $k \in \mathbb{N}$

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

assume $i \leq j_k \leq n$ and $\forall p \in \mathbb{N}$, $i \leq p \leq j_k - 1$, res $\leq$ distance(A[i-1], A[p]) is true.

We want to show that at the end of (k+1)-th iteration of inner loop, $\forall p \in \mathbb{N}$, $i \leq p \leq j_{k+1} - 1$, res $\leq$ distance(A[i-1], A[p]) is true.

Case 1: there is no (k+1)-th iteration of inner loop, then $j_{k+1} = j_k$, then $i \leq j_{k+1} \leq n$ and $\forall p \in \mathbb{N}$, $i \leq p \leq j_{k+1} - 1$, res $\leq$ distance(A[i-1], A[p]) is true.

Case 2: there is (k+1)-th iteration of inner loop, then $i \leq j_k < n$, then $j_{k+1} = j_k + 1$, then $i \leq j_{k+1} \leq n$.

Subcase 1: res updates because the condition of the if-statement in the inner loop is triggered, i.e., distance(A[i-1], A[j_{k+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[j_{k+1}-1]) $\geq$ dis.

By inductive hypothesis, $\forall p \in \mathbb{N}$, $i \leq p \leq j_k - 1$, res $\leq$ distance(A[i-1], A[p]), combining with the result from subcase 1 and subcase 2,

then $i \leq j_{k+1} \leq n$ and $\forall p \in \mathbb{N}$, $i \leq p \leq j_{k+1} - 1$, res $\leq$ distance(A[i-1], A[p]).

Prove for termination of inner loop:

Let $E_k = n - j_k$. By loop invariant, $j_k \leq n$. So $E_k \geq 0 \Longrightarrow E_k \in \mathbb{N}$, $\forall k$.

If there is a k+1 iteration,then $E_{k+1} = n - j_{k+1} = n - (j_k + 1) < n - j_k = E_k$.

Thus, the inner loop terminates.

Prove for outer loop's partial correctness.

Base Case:

Since, $i_0 = 0$

Then $i_0 \leq n$,

Since, $i_0 - 1 < 0$, then,

$\forall$ m $\in \mathbb{N}$, $0 \leq m \leq i-1$, $\forall s \in \mathbb{N}$, $0 \leq s \leq n-1$, $m \neq s \Longrightarrow res \leq distance(A[m], A[s])$ is vacuous true.

Inductive Steps:

Let $i_k \in \mathbb{N}$, where $k \in \mathbb{N}$.

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

assume that $i_k \leq n$ and

$\forall$ m $\in \mathbb{N}$, $0 \leq m \leq i_k - 1$, $\forall s \in \mathbb{N}$, $0 \leq s \leq n-1$, $m \neq s \Longrightarrow res \leq distance(A[m], A[s])$

WTS: at the end of (k+1)th iteration of outer loop, $i_{k+1} \leq n$ and $\forall$ m $\in \mathbb{N}$, $0 \leq m \leq i_{k+1} - 1$, $\forall s \in \mathbb{N}$, $0 \leq s \leq n-1$, $m \neq s \Longrightarrow res \leq distance(A[m], A[s])$

Case 1: there is no (k+1)th iteration of outer loop, then $i_{k+1} = i_k \leq n$ and

$\forall$ m $\in \mathbb{N}$, $0 \leq m \leq i_{k+1} - 1$, $\forall s \in \mathbb{N}$, $0 \leq s \leq n-1$, $m \neq s \Longrightarrow res \leq distance(A[m], A[s])$

(By Induction Hypothesis)

Case 2: there is (k+1)th iteration of outer loop, it means $i_k < n$, then $i_{k+1} \leq n$.

Assume that at the end of (k+1)-th iteration of outer loop, the post-condition of inner loop is correct, i.e., $\forall p \in \mathbb{N}$, $i_{k+1} \leq p \leq n-1$, $res \leq distance(A[i_{k+1} - 1], A[p])$.

Combining with Induction Hypothesis, we can get:

$\forall$ m $\in \mathbb{N}$, $0 \leq m \leq i_{k+1} - 1$, $\forall s \in \mathbb{N}$, $0 \leq s \leq n-1$, $m \neq s \Longrightarrow res \leq distance(A[m], A[s])$

Prove for outer loop's termination.

Let $E_k = n - i_k$. By loop invariant, $i_k \leq n$. So $E_k \geq 0 \Longrightarrow E_k \in \mathbb{N}$, $\forall k$.

If there is a k+1 iteration,then $E_{k+1} = n - i_{k+1} = n - (i_k + 1) < n - j_k = E_k$.

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.