My name and student number: Joseph Siu, 1010085701. Sanchit Manchanda, Ali Zaki Rashid.

1. Consider the following iterative algorithm that finds the length of the longest increasing subsequence in the array A[1..n].

```
1 L[1] ← 1
2 for i ← 2  to n do
3     L[i] ← 1
4     for j ← 1  to i − 1 do
5         if (A[j] < A[i]) and (L[j] ≥ L[i]) then  L[i] ← L[j] + 1
6 m ← L[n]
7 for i ← 2  to n − 1 do
8     if L[i] > m then  m ← L[i]
```

(a) Give a precise statement of what it means for this algorithm to be partially correct.

Precondition: n is a positive integer, and A[1..n] is an array with elements from a totally ordered domain.

Postcondition: The array A[1..n] is unchanged, and m is assigned with a positive integer which represents the length of the longest increasing subsequence in A[1..n].

Partially correct: If n is a positive integer n, A[1..n] is an array with elements from a totally ordered domain, and the algorithm is executed and terminated, then A[1..n] is unchanged, and m is assigned with a positive integer which represents the length of the longest increasing subsequence in A[1..n].

(b) Prove that this algorithm is partially correct.

We will use L1, L2 to denote line 1, line 2, and so on.

We also assumed the line numbers of the pseudo-code are fixed (L6, L7, L8 instead of L7, L8, L9).

Proof of Question 1(b).

Assume n is a positive integer, A[1..n] is an array with elements from a totally ordered domain, and the algorithm is executed and terminated.

Since there are no assignments to A[1..n] in the algorithm, A[1..n] is unchanged.

For $l \in \mathbb{N}$. Let $Q(l)$ = "immediately after the l^{th} iteration, $L[l+1]$ contains the length of the longest increasing (finite) subsequence in A[1..$(l+1)$] that ends with the term A[$l+1$]$''$; and let $P(l)$="If the for-loop from line 2 to line 5 is executed at least l times, then $Q(l)$."

Lemma 1. $\forall l \in \mathbb{N}.P(l)$.

Proof of Lemma 1 by strong induction.

Let $l \in \mathbb{N}$ be arbitrary;

Assume $\forall k \in \mathbb{N}.(k < l)$ IMPLIES $P(k)$.

Assume the for-loop from line 2 to line 5 is executed at least l times.

Case 1. $l = 0$.

Then trivially the length of the longest increasing subsequence in A[1..1] is 1, and we assigned $L[1] = 1$ on L1. Thus $Q(0)$ holds.

For Case 1 $Q(l)$.

Case 2. $l = 1$.

Then for the first iteration, $i = 2$; on L3 $L[2]$ is assigned with 1; on L4 since j is from 1 to $2 - 1 = 1$, we only execute L5 once where $j = 1$, and there are 2 subcases due to $L[1] = 1 \geq 1 = L[2]$:

Subcase (1): $A[j] < A[i]$

Then on L5 $L[2]$ is assigned with $L[1] + 1 = 2$, after this we end this iteration, and now $L[2] = 2$ is indeed the length of the longest increasing subsequence in A[1..2].

Thus $Q(l)$ holds for subcase 2.1.

Subcase (2): $A[j] \geq A[i]$.

Then no assignment on L5 has been made. After this we end this iteration, and now $L[2] = 1$ is indeed the length of the longest increasing subsequence in A[1..2] since $\mathrm{A}[j] = \mathrm{A}[1] \geq \mathrm{A}[2] = \mathrm{A}[i]$.

Thus $Q(l)$ holds for subcase 2.2.

For all subcases of Case 2 we have shown $Q(l)$ holds, so for Case 2 $Q(l)$.

Case 3. $l \geq 2$.

We first assign $L[l] = 1$ on L3.

Now, let $S = \{p \in [l] \mid A[p] < A[l]\}$, and let $S' = \{L[p] \mid p \in S\}$.

Subcase (1): $S = \emptyset$.

Then since for all $p \in [l].\mathrm{NOT}(A[p] < A[l])$, no assignment on L5 has been made, and $L[l+1] = 1$ is indeed the length of the longest increasing subsequence in $\mathrm{A}[1..(l+1)]$ with the last term $\mathrm{A}[l+1]$ (all previous terms are at least $\mathrm{A}[l+1]$).

Thus $Q(l)$ holds for subcase 3.1.

Subcase (2): $S \neq \emptyset$.

Then by construction this implies $S' \neq \emptyset$.

$i = l + 1$ on L2.

Since S' is a finite non-empty subset of $\mathbb{Z}^+$, we are allowed to construct $q = \max S'$.

Then $q \in S'$, by construction there exists $p' \in S$ such that $q = L[p']$, we instantiate such p'.

By inductive hypothesis (specialization and modus ponens), q is the length of the longest increasing subsequence in $\mathrm{A}[1..p']$ that ends with the term $\mathrm{A}[p']$. We instantiate such subsequence as $\{s_o\}_{o=1}^{q}$. Moreover, because $q = \max S'$, this means q is the length of the longest increasing subsequence in $\mathrm{A}[1..l]$ that ends with a term less than $\mathrm{A}[l]$.

Now, we claim the subsequence $\{s_o\}_{o=1}^{q} \circ \{\mathrm{A}[l+1]\}$ is the longest increasing subsequence in $\mathrm{A}[1..(l+1)]$ that ends with the term $\mathrm{A}[l+1]$. Indeed, firsly for a subsequence to be both increasing and ends with $\mathrm{A}[l+1]$, the term before $\mathrm{A}[l+1]$ must be less than $\mathrm{A}[l+1]$, and by A8 $\{s_o\}_{o=1}^{q} \circ \{\mathrm{A}[l+1]\}$ is increasing since $\{s_o\}_{o=1}^{q}$ is increasing and $s_q = \mathrm{A}[p'] < \mathrm{A}[l+1]$ by definition of S. Secondly, obviously the concatenation shows the subsequence ends with $\mathrm{A}[l+1]$. Lastly, $\{s_o\}_{o=1}^{q}$ being the longest increasing subsequence in $\mathrm{A}[1..l]$ implies $\{s_o\}_{o=1}^{q} \circ \{\mathrm{A}[l+1]\}$ is the longest increasing subsequence in $\mathrm{A}[1..(l+1)]$ that ends with the term $\mathrm{A}[l+1]$.

Therefore, as long as we show $L[l+1] = q + 1$, we can conclude $Q(l)$ holds.

To this end, since $1 \leq p' \leq l = i - 1$, for the for-loop from L4 to L5, right after the iteration where $j = p'$, since $A[p'] < A[l]$, and $L[p'] \geq L[l]$ due to our construction of q, on L5 $L[l+1]$ is assigned with $L[p'] + 1 = q + 1$. And after this iteration, consider 2 cases:

1. If $A[j] < A[l]$, then since $L[x] \leq L[p'] = L[l+1]$ for all $x \in S'$ by construction of q, no assignment on L5 has been made.

2. If $A[j] \geq A[l]$, then no assignment on L5 has been made.

For both subcases 3.2.1 and 3.2.2, we have shown that immediately after the l^{th} iteration, $L[l+1] = q + 1$ is indeed the length of the longest increasing subsequence in $\mathrm{A}[1..(l+1)]$

that ends with the term A$[l+1]$. Thus $Q(l)$ holds for subcase 3.2.

For all subcases of Case 3 we have shown $Q(l)$ holds, so for Case 3 $Q(l)$.

For all cases we have shown $Q(l)$ holds, so $Q(l)$.

By direct proof, $P(l)$.

By strong induction, $\forall l \in \mathbb{N}.P(l)$.

QUOD ERAT DEM■

Now, since there are no assignments to i and j within their for-loop respectively, and so only finitely many for-loop iterations has performed thus the for loop from L2 to L5 eventually terminates. Now we are on L6.

Case 1. $n = 1$.

Then we assign m with $L[n] = L[1] = 1$ and the algorithm terminates. Thus m is assigned with a positive integer which represents the length of the longest increasing subsequence in A$[1..n]$.

For Case 1 m is assigned with a positive integer which represents the length of the longest increasing subsequence in A$[1..n]$.

Case 2. $n \geq 2$.

From L6 to L8, we assign m with the largest value in $L[2..n]$, since $L[n] \geq L[1]$, such m is also the largest value in $L[1..n]$. Since by Lemma 1 $L[y]$ is the length of the longest increasing subsequence in A$[1..y]$ ending with A$[y]$ for all $y \in [n]$, so the maximum of $L[1..y]$ is indeed the length of the longest increasing subsequence in A$[1..n]$.

For Case 2 m is assigned with a positive integer which represents the length of the longest increasing subsequence in A$[1..n]$.

For all cases m is assigned with a positive integer which represents the length of the longest increasing subsequence in A$[1..n]$, thus by proof of conjunction A$[1..n]$ is unchanged, and m is assigned with a positive integer which represents the length of the longest increasing subsequence in A$[1..n]$.

By direct proof, the algorithm is partially correct.

QUOD ERAT DEM■

2. For each $k \in \mathbb{Z}^+$, let $X_k = \{x \in \{0,1\}^* : \text{NOT}(\exists y \in \{0,1\}^k.(x = y \cdot y))\}$
and consider the NFA $N_k = (Q, \{0,1\}, \delta, q_0, F)$, where:
$Q = \{q_i : 0 \leq i \leq 2k+1\} \cup \{p_i : 0 \leq i \leq k-1\} \cup \{z_i : 0 \leq i \leq k-1\}$,
$F = \{q_i : 0 \leq i \leq 2k-1\} \cup \{q_{2k+1}\}$,
$\delta(q_i, 0) = \{q_{i+1}, z_0\}$ for $0 \leq i \leq k-1$,
$\delta(q_i, 1) = \{q_{i+1}, p_0\}$ for $0 \leq i \leq k-1$,
$\delta(q_i, 0) = \delta(q_i, 1) = \{q_{i+1}\}$ for $k \leq i \leq 2k$,
$\delta(q_{2k+1}, 0) = \delta(q_{2k+1}, 1) = \{q_{2k+1}\}$,
$\delta(z_i, 0) = \delta(z_i, 1) = \{z_{i+1}\}$ for $0 \leq i \leq k-2$,
$\delta(z_{k-1}, 1) = \{q_{2k+1}\}$,
$\delta(z_{k-1}, 0) = \varnothing$,
$\delta(p_i, 0) = \delta(p_i, 1) = \{p_{i+1}\}$ for $0 \leq i \leq k-2$,
$\delta(p_{k-1}, 0) = \{q_{2k+1}\}$,
$\delta(p_{k-1}, 1) = \varnothing$, and
$\delta(q, \lambda) = \varnothing$ for all $q \in Q$.

(a) For each state $q \in Q$, describe the set of strings $w \in \{0,1\}^*$ such that $q \in \delta^*(q_0, w)$.
Your descriptions should not mention δ.

Description. We will use "letter" to denote a single element in $\Sigma = \{0,1\}$.

Let $k \in \mathbb{Z}^+$ be arbitrary. By definition of δ^*, it is equivalent to describe for each $q \in Q$, what strings $w \in \{0,1\}^*$ will lead to state q from the initial state q_0. We will prove our description in part (b).

Now, if $q = q_i$ for some $i \in [2k] \cup \{0\}$ (we instantiate such i), then q can be reached from q_0 by a string $w \in \{0,1\}^*$ if and only if w is a string with length i.

If $q = q_{2k+1}$, there are 2 cases that the string will reach q_{2k+1}: First, any string $w \in \{0,1\}^*$ with length at least $2k+1$ will reach q_{2k+1}; Second, if $w \in \{0,1\}^*$ has length at least $k+1$, and there exists 2 letters $a \in w$, $b \in w$ such that $a \neq b$ and they are $k-1$ letters apart, then w will reach q_{2k+1} (by "$\in$" we mean a is one of the letters of the string w, etc).

Let $\omega = \{w \in \{0,1\}^* \mid\ w \text{ has length at most } k-1\}$.

For p_0, if $w = w' \cdot 1$ for some $w' \in \omega$, then w will reach p_0; For p_i with $i \in [k-1]$, if $w = w' \cdot 1 \cdot w''$ for some $w' \in \omega$ and $w'' \in \{0,1\}^i$, then w will reach p_i.

Similarly, for z_0, if $w = w' \cdot 0$ for some $w' \in \omega$, then w will reach z_0; For z_i with $i \in [k-1]$, if $w = w' \cdot 0 \cdot w''$ for some $w' \in \omega$ and $w'' \in \{0,1\}^i$, then w will reach z_i.

(b) Prove that $L(N_k) = X_k$ for all $k \in \mathbb{Z}^+$.

Lemma 1. For all $k \in \mathbb{Z}^+$. For all $x \in \{0,1\}^* - X_k$. $|x| = 2k$.

Proof of Lemma 1.

Let $k \in \mathbb{Z}^+$ be arbitrary.

Let $x \in \{0,1\}^* - X_k$ be arbitrary.

Since $X_k \subseteq \{0,1\}^*$, by definition of X_k we have $\{0,1\}^* - X_k = \{x \in \{0,1\}^* : \exists y \in \{0,1\}^k.(x = y \cdot y)\}$, since $x \in \{0,1\}^* - X_k$, we instantiate $y \in \{0,1\}^k$ such that $x = y \cdot y$.

Since $|y| = k$ and $|y \cdot y| = 2k$, by substitution we have $|x| = 2k$.

Since x is arbitrary, we conclude for all $x \in \{0,1\}^* - X_k$. $|x| = 2k$.

Since k is arbitrary, we conclude Lemma 1 holds.

QUOD ERAT DEM■

Proof of Question 2(b).

Let $k \in \mathbb{Z}^+$ be arbitrary.

Let $w \in \{0,1\}^*$ be arbitrary.

Consider 3 cases of the length of w: $|w| < 2k, |w| = 2k, |w| > 2k$.

Since there are no λ transitions, for all $q \in Q$ we have $\delta(q, \lambda) = \varnothing$, so we will simply ignore checking λ transitions.

Case 1. $|w| < 2k$.

Since $\delta(q_i, 0) = \{q_{i+1}, z_0\}$ and $\delta(q_i, 1) = \{q_{i+1}, p_0\}$ for $0 \leq i \leq k-1$, and $\delta(q_i, 0) = \delta(q_i, 1) = \{q_{i+1}\}$ for $k \leq i \leq 2k$, from these we have $q_{i+1} \in \delta(q_i, \alpha)$ for $0 \leq i \leq 2k$ where $\alpha \in \{0,1\}$.

Hence, since q_i is a final state for $i \in [2k+1] \cup \{0\} - \{2k\}$, the string w starting from q_0 and through the walk $\{\omega_j\}_{j \in [|w|]}, \omega_j = (q_{j-1}, q_j)$ reaches a final state $q_{|w|}$ since $|w| < 2k$ and so $|w| \in [2k+1] \cup \{0\} - \{2k\}$ by our case assumption.

For Case 1 we have shown that when $|w| < 2k$, w is accepted by N_k, thus $w \in L(N_k)$. Moreover, by Lemma 1 and specialization of k, because $|w| \neq 2k$, thus $w \notin \{0,1\}^* - X_k$, so $w \in X_k$. Therefore $w \in L(N_k)$ if and only if $w \in X_k$.

Case 2. $|w| > 2k$.

$|w| > 2k$ implies $|w| \geq 2k+1$, so $|w| - 2k - 1 \geq 0$. We will use $[0]$ to denote the empty set.

Since $\delta(q_i, 0) = \{q_{i+1}, z_0\}$ and $\delta(q_i, 1) = \{q_{i+1}, p_0\}$ for $0 \leq i \leq k-1$, and $\delta(q_i, 0) = \delta(q_i, 1) = \{q_{i+1}\}$ for $k \leq i \leq 2k$, from these we have $q_{i+1} \in \delta(q_i, \alpha)$ for $0 \leq i \leq 2k$ where $\alpha \in \{0,1\}$.

Moreover, since $\delta(q_{2k+1}, 0) = \delta(q_{2k+1}, 1) = \{q_{2k+1}\}$, and q_{2k+1} is a final state, by constructing the walk $\{(q_{i-1}, q_i)\}_{i \in [2k+1]} \circ \{(p_j, p_j)\}_{j \in [|w|-2k-1]}, p_j = q_{2k+1}$ for all $j \in [|w| - 2k - 1]$. We can see that w is accepted by N_k since it reaches the final state q_{2k+1} through such walk.

For Case 2 we have shown that when $|w| > 2k$, w is accepted by N_k, thus $w \in L(N_k)$. Moreover, by Lemma 1 and specialization of k, because $|w| \neq 2k$, thus $w \notin \{0,1\}^* - X_k$, so $w \in X_k$. Therefore $w \in L(N_k)$ if and only if $w \in X_k$.

Case 3. $|w| = 2k$.

We will use the word "character" to denote element in $\Sigma = \{0,1\}$, moreover, since w is a finite sequence of characters, we will use w_i to denote the i^{th} character of w starting from position 1.

Subcase (1): $\text{NOT}(\exists y \in \{0,1\}^k.(w = y \cdot y))$.

First, $w = y \cdot y = yy$ is equivalent to $\forall i \in [2k].\ w_i = yy_i$.

Hence, $\text{NOT}(\exists y \in \{0,1\}^k.(w = y \cdot y))$ is equivalent to $\forall y \in \{0,1\}^k.\exists i \in [2k].w_i \neq yy_i$.

Let $y' \in \{0,1\}^k$ be such that $\forall i \in [k].y'_i = w_i$. Then by specialization of the above formula, we have $\exists i \in [2k].w_i \neq y'y'_i$.

Since by construction $\forall i \in [k].w_i = y'y'_i$, combining with $\exists i \in [2k].w_i \neq y'y'_i$ we have $\exists i \in [2k] - [k].w_i \neq y'y'_i$, instantiate such i.

Since $w_i \neq y'y'_i = y'y'_{i-k}$ but by specialization $w_{i-k} = y'_{i-k} = y'y'_{i-k}$, by substitution we have $w_i \neq w_{i-k}$.

Subsubcase (a) $w_i = 0$.

Since $w_i \neq w_{i-k}$, we have $w_{i-k} = 1$.

Now define $\forall m \in [2k-i], r_m = (q_{2k+1}, q_{2k+1})$, and construct the walk $\{(q_{j-1}, q_j)\}_{j \in [i-k-1]} \circ \{(q_{i-k-1}, p_0)\} \circ \{(p_{l-1}, p_l)\}_{l \in [k-1]} \circ \{(p_{k-1}, q_{2k+1})\} \circ \{r_m\}_{m \in [2k-i]}$. Here

$\{(q_{j-1}, q_j)\}_{j \in [i-k-1]}$ is valid since $q_{b+1} \in \delta(q_b, 0) = \delta(q_b, 1)$ for $0 \leq b \leq 2k$ from Case 2;

$\{(q_{i-k-1}, p_0)\}$ is valid due to $\delta(q_a, 1) = \{q_{a+1}, p_0\}$ for $0 \leq a \leq k-1$ and $w_{i-k} = 1$;

$\{(p_{l-1}, p_l)\}$ is valid due to $\delta(p_c, 0) = \delta(p_c, 1) = \{p_{c+1}\}$ for $0 \leq c \leq k-2$;

$\{(p_{k-1}, q_{2k+1})\}$ is valid due to $\delta(p_{k-1}, 0) = \{q_{2k+1}\}$ and $w_i = 0$ (the length of the walk before is $(i-k-1) + (1) + (k-1) = i-1$, thus we read the character w_i during $\{(p_{k-1}, q_{2k+1})\}$);

and finally $\{r_m\}_{m \in [2k-i]}$ is valid due to $\delta(q_{2k+1}, 0) = \delta(q_{2k+1}, 1) = \{q_{2k+1}\}$.

So, we have shown our walk is valid and reaches the final state q_{2k+1}. Note that all indexes are valid due to basic algebra and arithmetic manipulations.

For subsubcase (a) we have shown w is accepted by N_k thus $w \in L(N_k)$. And by definition of X_k, $w \in X_k$. So for subsubcase (a) $w \in L(N_k)$ if and only if $w \in X_k$.

Subsubcase (b) $w_i = 1$.

Since $w_i \neq w_{i-k}$, we have $w_{i-k} = 0$.

Now define $\forall m \in [2k-i], r_m = (q_{2k+1}, q_{2k+1})$, and construct the walk $\{(q_{j-1}, q_j)\}_{j \in [i-k-1]} \circ \{(q_{i-k-1}, z_0)\} \circ \{(z_{l-1}, z_l)\}_{l \in [k-1]} \circ \{(z_{k-1}, q_{2k+1})\} \circ \{r_m\}_{m \in [2k-i]}$. Here

$\{(q_{j-1}, q_j)\}_{j \in [i-k-1]}$ is valid since $q_{b+1} \in \delta(q_b, 0) = \delta(q_b, 1)$ for $0 \leq b \leq 2k$ from Case 2;

$\{(q_{i-k-1}, z_0)\}$ is valid due to $\delta(q_a, 0) = \{q_{a+1}, z_0\}$ for $0 \leq a \leq k-1$ and $w_{i-k} = 0$;

$\{(z_{l-1}, z_l)\}$ is valid due to $\delta(z_c, 0) = \delta(z_c, 1) = \{z_{c+1}\}$ for $0 \leq c \leq k-2$;

$\{(z_{k-1}, q_{2k+1})\}$ is valid due to $\delta(z_{k-1}, 1) = \{q_{2k+1}\}$ and $w_i = 1$ (the length of the walk before is $(i-k-1) + (1) + (k-1) = i-1$, thus we read the character w_i during $\{(z_{k-1}, q_{2k+1})\}$);

and finally $\{r_m\}_{m \in [2k-i]}$ is valid due to $\delta(q_{2k+1}, 0) = \delta(q_{2k+1}, 1) = \{q_{2k+1}\}$.

So, we have shown our walk is valid and reaches the final state q_{2k+1}. Note that all indexes are valid due to basic algebra and arithmetic manipulations.

For subsubcase (b) we have shown w is accepted by N_k thus $w \in L(N_k)$. And by definition of X_k, $w \in X_k$. So for subsubcase (b) $w \in L(N_k)$ if and only if $w \in X_k$.

For subcase (1) we have shown $w \in L(N_k)$ and $w \in X_k$. Thus $w \in L(N_k)$ if and only if $w \in X_k$.

Subcase (2): $\exists y \in \{0,1\}^k.(w = y \cdot y)$.

Since $\exists y \in \{0,1\}^k.(w = y \cdot y)$, we instantiate such y.

By definition of X_k, $w \notin X_k$. We now show $w \notin L(N_k)$.

To this end, since $|w| = 2k$, and the length of path to read q_d where $d \in [2k-1] \cup \{0\}$ is exactly d by our definitions of δ: q_d can only be reached from q_{d-1} for $d \in [2k-1]$ and q_0 can only be reached only by the string λ, solving the simple recurrence relation we have q_b can only be reached by the strings with length b for all $b \in [2k-1] \cup \{0\}$.

So, since $|w| = 2k > 2k-1$, final states $\{q_d \mid d \in [2k-1] \cup \{0\}\}$ cannot be reached by w, now we just have to show q_{2k+1} cannot be reached by w.

To obtain a contradiction, assume there exists a walk that q_{2k+1} is reached by w from q_0.

There are 3 possible paths to reach q_{2k+1}: from q_{2k}, from p_{k-1}, and from z_{k-1}.

Subsubcase (a) w can be reached from q_{2k}.

For this case since to reach q_{2k-1}, the path must be at least length $2k-1$, and to reach q_{2k}, the path must be at least length $2k$, so the path must be at least length $2k+1$ to reach q_{2k+1}, which is a contradiction to $|w| = 2k$.

For this subsubcase (a) contradiction occured.

Subsubcase (b) w can be reached from p_{k-1}.

Since $q_{2k+1} \in \delta(p_{k-1}, 0)$ and $\delta(p_{k-1}, 1) = \varnothing$, this implies $w_i = 0$ for some $i \in [2k-1] \cup \{0\}$, since the shortest path to reach p_{k-1} is length k (from q_0 to p_0 and from p_0 to p_{k-1}), this gives $k+1 \leq i \leq 2k-1$. Since $w = kk$ and $w_i = 0$, this shows $w_{i-k} = 0$ (valid index since $i \geq k+1$). However, since the path from p_0 to p_{k-1} is exactly length k, and p_0 can only be reached by character 1, this implies $w_{i-k} = 1$, which is a contradiction.

For this subsubcase (b) contradiction occured.

Subsubcase (c) w can be reached from z_{k-1}.

Since $q_{2k+1} \in \delta(z_{k-1}, 1)$ and $\delta(z_{k-1}, 0) = \varnothing$, this implies $w_i = 1$ for some $i \in [2k-1] \cup \{0\}$, since the shortest path to reach z_{k-1} is length k (from q_0 to z_0 and from z_0 to z_{k-1}), this gives $k+1 \leq i \leq 2k-1$. Since $w = kk$ and $w_i = 1$, this shows $w_{i-k} = 1$. However, since the path from z_0 to z_{k-1} is exactly length k, and z_0 can only be reached by character 0, this implies $w_{i-k} = 0$, which is a contradiction.

For this subsubcase (c) contradiction occured.

For all cases contradiction occured, so q_{2k+1} cannot be reached by w. Since w cannot reach all final states, we conclude $w \notin L(N_k)$, and by definition of X_k, $w \notin X_k$. Therefore $w \in L(N_k)$ if and only if $w \in X_k$.

For subcase (2) we have shown $w \notin L(N_k)$ and $w \notin X_k$. Thus $w \in L(N_k)$ if and only if $w \in X_k$.

We conclude for Case 3 $w \in L(N_k)$ if and only if $w \in X_k$.

For all cases we have shown $w \in L(N_k)$ if and only if $w \in X_k$.

Since w is arbitrary, we conclude $\forall w \in \{0,1\}^*.w \in L(N_k)$ if and only if $w \in X_k$. Since $L(N_k) \subseteq \{0,1\}^*$ and $X_k \subseteq \{0,1\}^*$, by definition of set equality we conclude $L(N_k) = X_k$.

Since k is arbitrary, we therefore have shown that $L(N_k) = X_k$ for all $k \in \mathbb{Z}^+$.

QUOD ERAT DEM■