Name: Harshit Jain
User ID: hmj5262

Problem 1

1. Let's suppose $\sqrt{13}$ is a rational number. Then we can write it $\sqrt{13} = \frac{a}{b}$ where a, b are whole numbers, b not zero.

   We additionally assume that this $\frac{a}{b}$ is simplified to lowest terms, since that can obviously be done with any fraction. Notice that in order for $\frac{a}{b}$ to be in simplest terms, both of a and b cannot be even. One or both must be odd. Otherwise, we could simplify $\frac{a}{b}$ further.

   From the equality $\sqrt{13} = \frac{a}{b}$ it follows that $13 = \frac{a^2}{b^2}$, or $a^2 = 13b^2$. Since 13 is prime and a^2 is a multiple of 13, then a is multiple of 13.

   If we substitute $a = 13k$ into the original equation $\sqrt{13} = \frac{a}{b}$, we get:

   $\Rightarrow (13k)^2 = 13b^2$

   $\Rightarrow b^2 = 13k^2$

   Since 13 is prime and b^2 is a multiple of 13 then b is multiple of 13.

   We now have a contradiction since a and b must have no common factors (except 1) but we have proved that if $\frac{a}{b}$ exits then a and b must have common factor 13.

   So $\frac{a}{b}$ can not exist and the square root of 13 is irrational.

2. Yes, we can prove that square root of any prime number is irrational.

   Let's suppose $\sqrt{p}$ is a rational number, where p is any prime number. Let $\sqrt{p} = \frac{m}{n}$ where $m, n \in N$. and m and n have no factors in common.

   Now $p = \frac{m^2}{n^2}$, or $m^2 = pn^2$.

   Since p is prime and m^2 is a multiple of p then m is multiple of p.

   If we substitute $m = pk$ into the original equation $\sqrt{p} = \frac{m}{n}$, we get:

   $\Rightarrow (pk)^2 = pn^2$

   $\Rightarrow n^2 = pk^2$

   Since p is prime and n^2 is a multiple of p then n is multiple of p.

   We now have a contradiction since m and n must have no common factors (except 1) but we have proved that if $\frac{m}{n}$ exits then m and n must have common factor p.

   So $\frac{m}{n}$ can not exist and the square root of any prime is irrational.

Problem 2

$Q = \{q_0, q_1, q_2, q_3\}$

q_0 : Initial state, state after reading a digit that leaves a remainder of 0 when divided by 4
q_1 : State after reading a digit that leaves a remainder of 1 when divided by 4
q_2 : State after reading a digit that leaves a remainder of 2 when divided by 4
q_3 : State after reading a digit that leaves a remainder of 3 when divided by 4

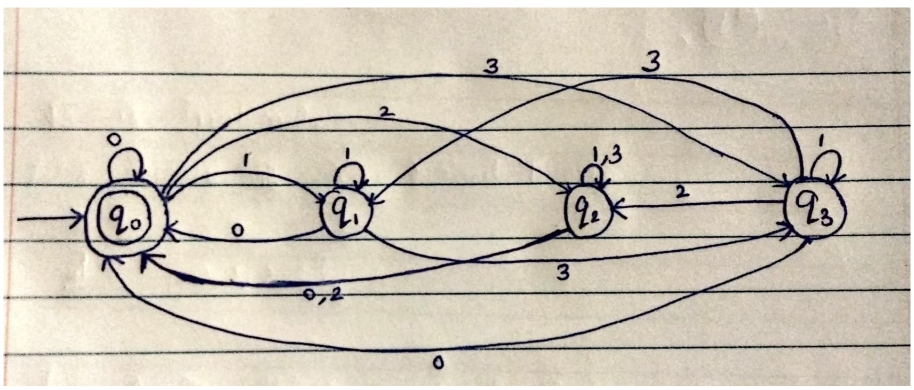
$\Sigma = \{0, 1, 2, 3\}$

$q_0 = \{q_0\}$ (Start State)

$F = \{q_0\}$ (Accept State)

$\delta =$

	0	1	2	3
q_0	q_0	q_1	q_2	q_3
q_1	q_0	q_1	q_2	q_3
q_2	q_0	q_2	q_0	q_2
q_3	q_0	q_3	q_2	q_1



Problem 3

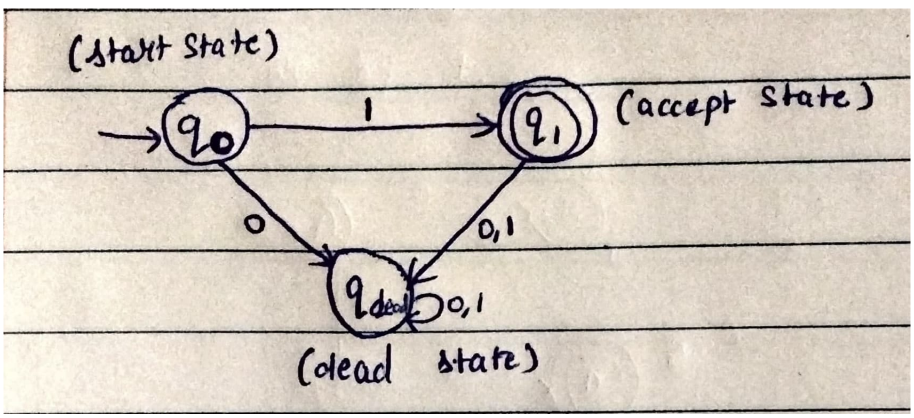
1. Assume there exists a DFA over the alphabet $\{0, 1\}$ that recognizes the language $L = \{1\}$ with less than three states.

   A DFA has states, transitions, an initial state, and accepting states. Let's analyze the possibilities:

   - We must have at least one start state. Let the start state be q_0.
   - For the string 1, there must be an accepting state. Let this state be q_1.
   - Let's introduce a dead state to q_{dead} handle strings other than 1. This state will be a non-accepting state, and we can transition to it from q_0 to q_1 on input 0 or 1.

   Now, this DFA has three states: q_0, q_1, and q_{dead}. It ensures that only the string "1" is accepted, and all other strings lead to the dead state.

   Therefore, we have shown that it is impossible to create a DFA with less than three states that recognizes the language $L = \{1\}$, and we need at least three states, including a dead state, to handle strings other than 1.

2. Yes, the DFA over the alphabet $\{1\}$ that recognizes the language $L = \{1\}$ needs to have at least three states.

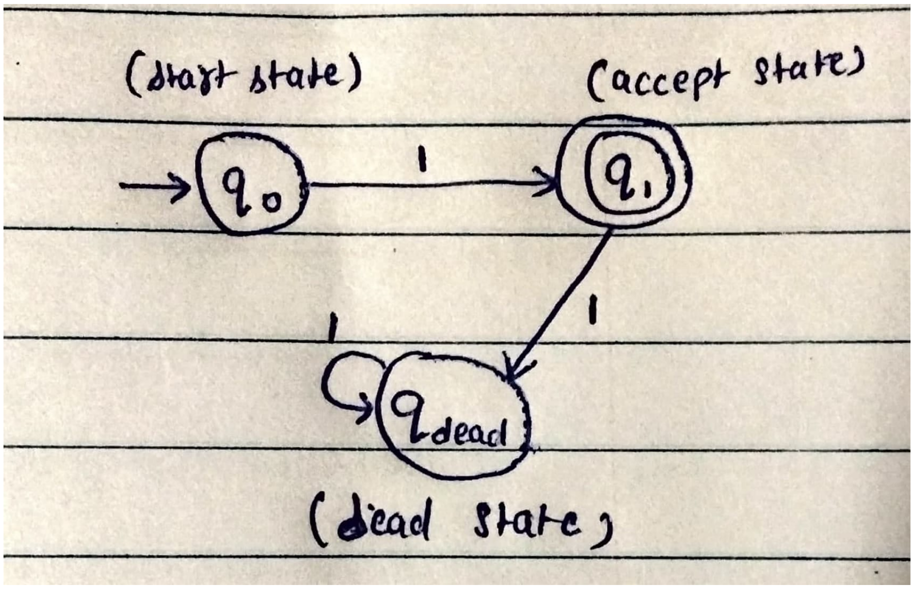
   <u>States:</u>

   - q_0 (Start state)
   - q_1 (Accept state for the valid string 1)
   - q_{dead} (Dead state for any other invalid string)

   <u>Transitions:</u>

   - From q_0, on input 1, transition to q_1
   - From q_1, on input 1, transition to q_{dead}

This DFA ensures that only the string 1 is accepted (transitioning from q_0 to q_1), and any additional 1s are rejected by transitioning to the dead state q_{dead}. Thus, a minimum of three states is required to handle both acceptance and rejection of strings in the language $L = \{1\}$ over the alphabet $\{1\}$.



Problem 4

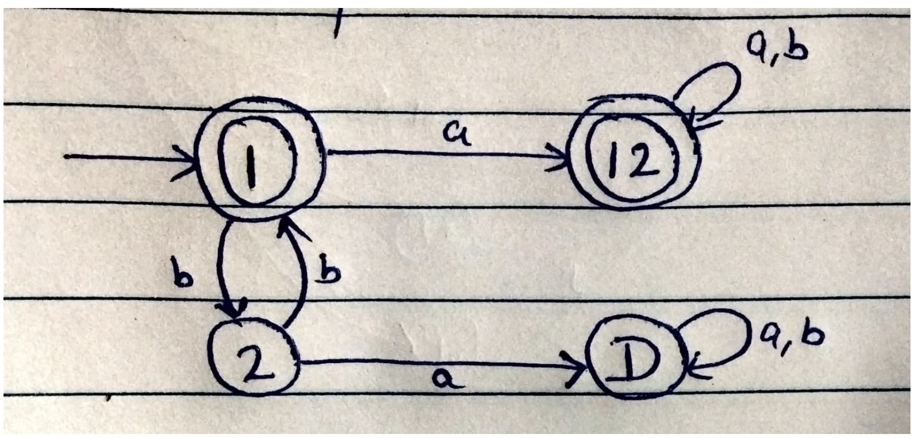
From the given NFA, we will make the transition table, $\delta_{NFA} = \begin{array}{c|cc} & \text{a} & \text{b} \\ \hline 1 & \{1,2\} & 2 \\ 2 & \phi & 1 \end{array}$

From here, we can make an equivalent DFA transition table as follows, $\delta_{DFA} = \begin{array}{c|cc} & \text{a} & \text{b} \\ \hline 1 & 12 & 2 \\ 12 & 12 & 12 \\ 2 & D & 1 \\ D & D & D \end{array}$

<u>States:</u>

- 1 (Start state and Accept State)
- 12 (Accept state)
- *D* (Dead state for any other invalid string)

So, the converted DFA is as follows:



Problem 5

1.
2.
3.