1. NFAs
(a) What language does the following NFA accept?

Solution:
All strings of only 0’s and 1’s not containing more than one 1.

(b) Create an NFA for the language “all binary strings that have a 1 as one of the last three digits”.

Solution:
The following is one such NFA:

2. DFAs & Minimization
(a) Convert the NFA from 1a to a DFA, then minimize it.

Solution:
Here is the minimized form:
(b) Minimize the following DFA:

Solution:

Step 1: $q_0, q_2$ are final states and the rest are not final. So, we start with the initial partition with the following groups: group 1 is \{q_0, q_2\} and group 2 is \{q_1, q_3, q_4\}.

Step 2: $q_1$ is sending $a$ to group 1 while $q_3, q_4$ are sending $a$ to group 2. So, we divide group 2. We get the following groups: group 1 is \{q_0, q_2\}, group 3 is \{q_1\} and group 4 is \{q_3, q_4\}.

Step 3: $q_0$ is sending $a$ to group 3 and $q_2$ is sending $a$ to group 4. So, we divide group 1. We will have the following groups: group 3 is \{q_1\}, group 4 is \{q_3, q_4\}, group 5 is \{q_0\} and group 6 is \{q_2\}.

The minimized DFA is the following:

3. RegExp to NFA

Use our generic construction to build an NFA that recognizes the language given by the following regular expression: $((0 \cup 1)^{*}001$. If you have time, also give as small an NFA as you can. (Unlike with DFAs there is no good minimization algorithm known for NFAs.)

Solution: