Math 3502: Combinatorial Optimisation. Solutions 1.

1. It is easier to check if a binary number is divisible by 4 than if it is divisible by 3. For the latter you actually have to do some dividing, for the former just check that the two lowest bits (the binary digits at the right hand end of the number) are equal to 0.

The following Turing machine does this (you could devise other ones). The head moves to the right until it reaches the end of the binary number.Then moves back leftwards checking whether the first two binary digits it meets are both 0. If they are it halts in state q_y and otherwise halts in state q_n.

$(q_0, *, q_n, *, \longrightarrow)$
$(q_0, 0, q_n, 0, \longrightarrow)$
$(q_0, 1, q_1, 1, \longrightarrow)$
$(q_1, 0, q_1, 0, \longrightarrow)$
$(q_1, 1, q_1, 1, \longrightarrow)$
$(q_1, *, q_2, *, \longleftarrow)$
$(q_2, 1, q_n, 1, \longleftarrow)$
$(q_2, 0, q_3, 0, \longleftarrow)$
$(q_3, 1, q_n, 1, \longleftarrow)$
$(q_3, 0, q_y, 0, \longleftarrow)$
$(q_3, *, q_n, *, \longleftarrow)$

2. Here we simply count the number of 1s we meet modulo 4. (So the version for 3 is conceptually the same and practically slightly faster to write down.) We use the condensed notation as in Talbot and Welsh. The machine halts in state q_{y_i} if the number of 1s in the binary number is equal to i mod 4.

$(q_0, *, q_{y_0}, *, \longrightarrow)$
$(q_0, 0, q_4, 0, \longrightarrow)$
$(q_0, 1, q_1, 1, \longrightarrow)$
$(q_1/q_2/q_3/q_4, 0, s, 0, \longrightarrow)$
$(q_1, 1, q_2, 1, \longrightarrow)$
$(q_2, 1, q_3, 1, \longrightarrow)$
$(q_3, 1, q_4, 1, \longrightarrow)$
$(q_4, 1, q_1, 1, \longrightarrow)$
$(q_1, *, q_{y_1}, *, \longrightarrow)$
$(q_2, *, q_{y_2}, *, \longrightarrow)$
$(q_3, *, q_{y_3}, *, \longrightarrow)$
$(q_4, *, q_{y_0}, *, \longrightarrow)$