605.621 Foundations of Algorithms Spring 2025
Programming Assignment #2
Assigned with Module 5, Due at the end of Module 7 (March 10)

The goals of Programming Assignment 2 are: (1) simulate a deterministic Turing machine (DTM). (2) apply your simulated DTM to four problems. Consult Course Modules |General Algorithm Links and/or |Advanced Algorithm Links, and our text if you find notation you are unfamiliar with in this assignment or in the CLRS book.

This assignment requires you to "construct an algorithm". There are, of course, many algorithms solving this problem. Some algorithms are asymptotically more efficient than others. Your objectives are to (a) complete the assignment, and (b) submit an asymptotically efficient algorithm. You are not beeing asked to find the most asymptotically efficient algorithm.

In this course, critical thinking and problem analysis results in discovering appropriate, and possible better or best, algorithm for a problem; defining the algorithm in pseudocode; demonstrating (we don't usually prove) that the algorithm is correct; and determining the asymptotic runtime for the algorithm using one or more of the tools for asymptotic analysis you have studied this semester. This programming problem is not a collaborative assignment.

Each algorithm you design must follow the **Programming Assignment Guidelines & Pseudocode Restrictions** on Canvas page *Programming Assignment Guidelines including Pseudocode Restrictions*)when preparing your solutions to these problems.

Exercising the algorithm you design using one or more example data set(s) removes any doubt from the grader that the algorithm is structurally correct and all computations are correct. When a problem supplies data you are expected to use it to demonstrate that your algorithm is correct.

You are permitted to use Internet resources while solving problems, however complete references must be provided. Please follow the Sheridan Libraries' citation guidance at "*Citing Other Things* - HOW DO YOU CITE AN INTERVIEW, A TWEET, OR A PUBLIC WEB PAGE?" and continue to the APA Academic Writer *Sample References*. Additional example citations are provided at the Purdue University, Purdue Online Writing Lab (OWL), College of Liberal Arts *Reference List: Electronic Sources*. You can also use training provided by Victoria University Library, Melbourne Australia on *Oxford Referencing: Internet/websites*

1. **Definition of Deterministic Turing Machines (DTM)**

   To begin, review the lecture "An Example DTM" in Module 3 Content.

   A Turing machine consists of the following:

   - A finite state control,
   - A tape consisting of a two-way sequence of tape squares, and
   - A read-write head.

   A DTM is defined by the content of each of the six following components:

   - A finite set Γ of symbols,
   - A finite set Q of states (q_0 is designated as the start state and q_Y and q_N are designated as halting states), where $\{q_0, q_H\} \in Q$,
   - The direction. $s =$ (left, right, halt), where left = -1 and right = +1, in which to move the tape head $s \in Q$
   - A transition function $\delta : (Q - \{q_Y, q_N\}) \times \Gamma \rightarrow Q \times \{+1, 0, -1\}$,
   - $\Sigma \subset \Gamma$ as a set of input symbols, and
   - $b \in \Gamma - \Sigma$ corresponds to the "blank symbol" (#, for example)

   This 6-tuple $M = (\Gamma, Q, s, \delta, \Sigma, b)$ and its content uniquely defines a DTM.

   The DTM will scan the tape, and the transition function will read the symbol from Γ currently written on the table and determine what character to write in its place as well as which direction s to move the read-write head. If the direction indicates $+1$, the head moves to the right, if the direction indicates -1, the head moves to the left and if the drection indicates 0, the head stays still.

   To see how a DTM works, suppose we have $\Sigma \subset \Gamma$ as a set of input symbols and $b \in \Gamma - \Sigma$ a "blank symbol" (#). Next suppose we have an input string $x \in \Sigma^*$ where we place the symbols of x in consecutive cells of the tape in positions $1...|x|$. All other cells on the tape are assumed to have b. The DTM will begin in state q_0 with the read-write head scanning square 1 and proceed according to the transition function.

   Assignment continues on the next page ...

If the current state of the DTM is ever q_Y or q_N, then the DTM terminates. On the other hand, if the current state is in $Q - \{q_Y, q_N\}$, then the DTM continues. When transitioning, the read-write head will first erase the symbol in the square it is examining and then write the new symbol as specified by the transition function. The read-write head then either moves one position to the right or one position to the left, and the Finite State Control updates the state to some successor q'.

Because we have limited the set of halting (or terminal) states to q_Y, q_N, we note that a DTM is only able to solve problems that return a Boolean result. In particular, we say that a DTM is used to solve decision problems where q_Y indicates the DTM has decided *yes* and q_N indicates the DTM has decided *no*.

2. **Module 3 Content - An Example DTM Explored**

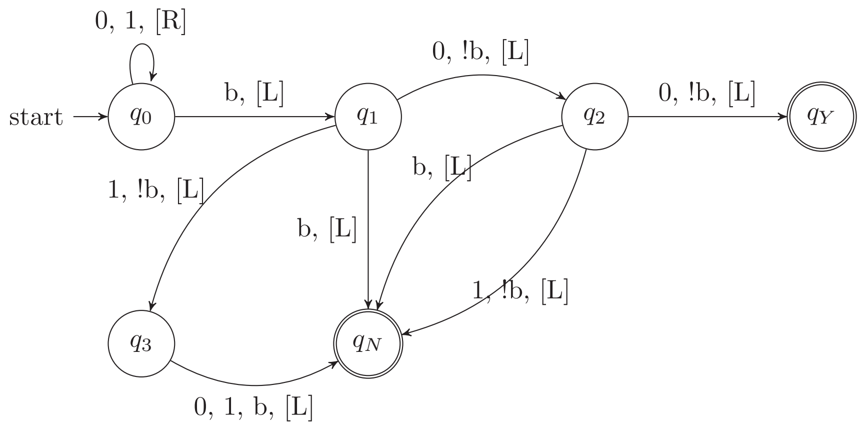

Figure 1: DTM State Diagram

The set of states is defined to be $Q = q_0, q_1, q_2, q_3, q_Y, q_N$, and the transition function δ is defined by the following table:

Table 1: DTM model illustrated in Figure 1.

$Q - \{q_Y, q_N\}$	0	1	b
q_0	$(q_0, 0, +1)$	$(q_0, 1, +1)$	$(q_1, b, -1)$
q_1	$(q_2, b, -1)$	$(q_3, b, -1)$	$(q_N, b, -1)$
q_2	$(q_Y, b, -1)$	$(q_N, b, -1)$	$(q_N, b, -1)$
q_3	$(q_N, b, -1)$	$(q_N, b, -1)$	$(q_N, b, -1)$

To reiterate how the DTM works, Table 1 represents the states $q = Q - \{q_Y, q_N\}$, symbols in x, and the direction the read-write head moves (s). Starting with the initial state represented with q_0 the finite controller reads the initial symbol represented by x. At each step the controller reads the symbol x on the tape at state q, enters the state $q' = Q = q_0, q_1, q_2, q_3, q_Y, q_N$, writes the symbol x' in the current cell, erasing x, and moves in the direction identified by s. This is represented as the five-tuple (q, x, q', x', s). For Table 1 the DTM is represented by the following finite-state machine in Figure 1. The edges in Figure 1 are labeled with the symbol(s) on the tape, a symbol that overwrites the current symbol (if any), and the directon of tape movement. For example, (1, !b, [L]) on the edge between q_1 and q_3 denotes when the read/write head is over a symbol 1 while in state q_1 overwrite the 1 with b and move the tape to the left.

Please note that you can define other symbols in Γ and you should. If you include S,E in your set Γ you could use those symbols to recognise the start of data on the tape (S) and the end of data on the tape (E) making those symbols available to you in your state modeling. Remember that each state must fully define the operation of the DTM in that state for each symbol in Γ.

Assignment continues on the next page ...

3. **PA2 Assignment**

   (a) [55 points] Implement the DTM example provided in the Course Content under Module 3, and described above including Figure 1 and Table 1 with your simulated DTM. Ensure that your 6-tuple, M, as listed above are clearly identified in your implementation to facilitate grading. Also implement the following capability in your simulated DTM :

      i. Print the state of the tape after each transition. If the number of transitions exceeds 30 also write the state of the tape to a file.

   (b) [45 points]Expand the implementation of the DTM simulator in part (a) to perform additional binary operations (addition, subtraction, multiplication

      i. [15 points] **Binary Addition -** The tape will need to handle variable length binary inputs, e.g., tape values of 101100101 add 101 or add 101101101.

Table 2: Binary Addition Rules

Operation $x + x'$	Sum	Carry
$0 + 0$	0	0
$0 + 1$	1	0
$1 + 0$	1	0
$1 + 1$	0	1

      ii. [15 points] **Binary Subtraction -** The tape will need to handle variable length binary inputs, e.g., tape values of 101100101 subtract 101 or subtract 101101.

Table 3: Binary Subtraction Rules

Operation $x - x'$	Sum	Carry
$0 - 0$	0	0
$0 - 1$	0	1
$1 - 0$	1	0
$1 - 1$	0	0

      iii. [**EXTRA CREDIT 15 points**] **Binary Multiplication -** The tape will need to handle variable length binary inputs, e.g., tape values of 101100101 multiply 101 or multiply 1011.

Table 4: Binary Multiplication Rules

Operation $x \times x'$	Multiplication
0×0	0
0×1	0
1×0	0
1×1	1

4. **PA2 Submission**

   (a) Everything required by Programming Assignment Guidelines, Section 1 Programming Assignment Summary, except pseudocode, asymptotic analysis of psudocode, and comparison of code work to algorithm asymptotic family.

   (b) State diagrams, as in Figure 1, may be carefully drawn by hand or an app and scanned with a flatbed scanner/multifunction printer. Illegible state diagrams will not be graded.