CS2351 Data Structures

Homework 3 (Suggested Solution)

There are many different ways to answer the questions perfectly. Don't worry if your answer is different with ours. In general, what we want to have is precise and concise.

1. **Ans:**

   (a) See Figure 1.

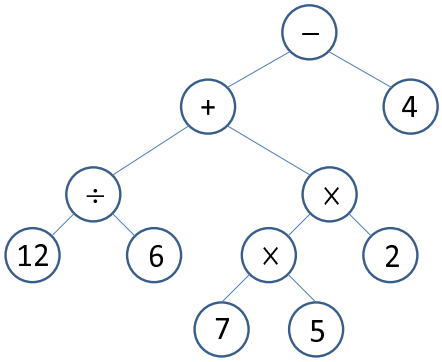

Figure 1: Expression tree for Q1(a)

   (b) Prefix notation:

$$- \quad + \quad \div \quad 12 \quad 6 \quad \times \quad \times \quad 7 \quad 5 \quad 2 \quad 4$$

   Postfix notation:

$$12 \quad 6 \quad \div \quad 7 \quad 5 \quad \times \quad 2 \quad \times \quad + \quad 4 \quad -$$

   (c) See Figure 2.

2. **Ans:**

   The following is a general algorithm that converts infix to postfix, when the expression does not contain parentheses:

```
1: Scan the infix expression from left to right ;
2: x = the operator or the number scanned ;
3: if x is a number then
4:    Output x immediately ;
5: else
6:    while stack is not empty AND top of stack has higher priority than x do
7:       y = top of stack ;
8:       Pop y from the stack, and output y immediately ;
9:    end while
10:   Push x to the stack ;
11: end if
12: Repeat Line 2 to Line 11 until all the expression is scanned ;
13: Pop and output all the operators remaining in the stack ;
```

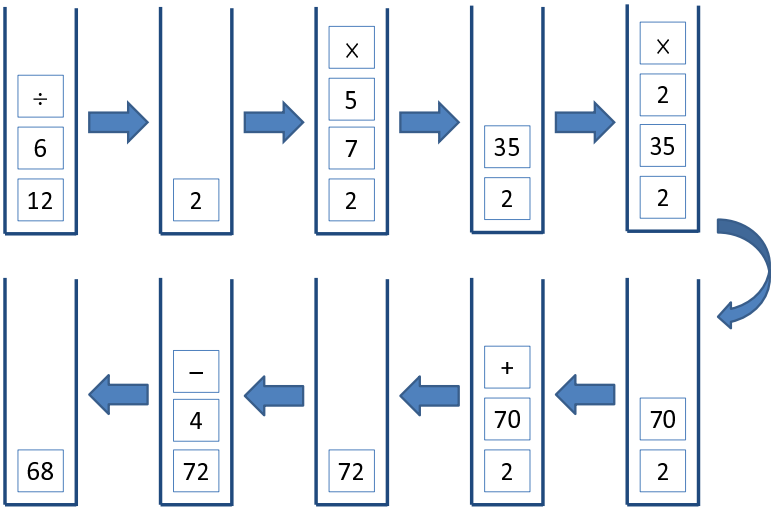

Figure 2: Key steps of evaluating the expression of Q1 in postfix notation

Here, the priority of the operators are defined as follows:

$$\circ > \circ_s > \triangle_s > \triangle > \times_s = \div_s > \times = \div > +_s = -_s > + = -,$$

where $\texttt{OP}_s$ indicates the case when the operator OP is in the stack.

3. **Ans:**

   (a) $M^2[u, v]$ counts the number of different ways to reach v from u in exactly 2 steps.

   (b) One way is to compute M^3 and then check for each u whether $M^3[u, u]$ is non-zero. Note that the value of $M^3[u, u]$ indicates the number of ways to reach u from u in exactly 3 steps, so that it is non-zero if and only if there is a triangle starting at u. Thus, the original graph has a triangle if and only if some entry $M^3[u, u]$ is non-zero. However, the computation of M^3 will require 2 matrix multiplications. We can in fact solve the problem just by comparing M^2 and M (thus using only one matrix multiplication). The idea is: For each entry $M^2[u, v]$ which is non-zero, we check if the entry $M[v, u]$ is 1. If such a pair (u, v) is found, then we have discovered a way to go from u to v in 2 steps, and then back to u in 1 more step. This implies that a triangle exists. Otherwise, if no such pair (u, v) is found, we can conclude that the graph does not contain any triangle.