Department of Computer Science and Engg, IIT Bombay

CS 213 (m): Data Structures and Algorithms Midsem: 17 Feb 2010; 14:00–16:00

One A4 Size Sheet of handwritten notes permitted. No Books or Photocopies. Weightage: 30%, Max Marks: 50

In case of any doubts in any question, make suitable assumptions, state them, justify them and proceed.

---

1. Given the following sequence of letters and asterisks: EAS*Y*QUE***ST***IO*N***

   (a) Consider the stack data structure, supporting two operations *push* and *pop*, as discussed in class. Suppose that for the above sequence, each letter (such as E) corresponds to a *push* of that letter onto the stack and each asterisk (*) corresponds a *pop* operation on the stack. Show the sequence of values returned by the *pop* operations. (**3 Marks**)

   (b) Consider the queue data structure, supporting two operations *insert* and *remove*, as discussed in class. Suppose that for the above sequence, each letter (such as E) corresponds to an *insert* of that letter into the queue and each asterisk (*) corresponds a *remove* operation on the queue. Show the sequence of values returned by the *remove* operations. (**2 Marks**)

2. An array of 10 elements, a[0:9], having values [4, 2, 6, 7, 1, 0, 9, 8, 5, 3] is to be sorted using *insertion sort*, as enacted in class.

   (a) Draw a figure to show the progress of the sorting, for the above sequence of input values. (**2 Marks**)

   (b) Assume that a *read* operation takes 1 unit of time, while a *write* operation takes 2 units of time. What is the amount of time taken to complete the sorting? Explain your reasoning used to compute the time taken. Also, explicitly mention any other assumptions that you may need. (**3 Marks**)

3. (a) Determine and explain the functionality of the following code fragment: (**2 Marks**)

```
int function1 (int a[], int n, int x)
{
    int i;
    for (i = 0; i < n && a[i] != x; i++);
    if (i == n) return -1;
    else return i;
}
```

   (b) Compute the best case, worst case and average case time complexity of the above function1. Explain your answers. (**3 Marks**)

   (c) Determine and explain the functionality of the following code fragment: (**2 Marks**)

```
int function2 (int a[], int n, int x)
{
    int i, j, k;
    i = 0; j = n -1;
    while (i <= j) {
        k = (i + j)/2;
        if (x == a[k]), return k;
        if (x > a[k]), i = k + 1;
        else j = k - 1;
    }
    return -1;
}
```

   (d) Compute the best case, worst case and average case time complexity of the above function2. Explain your answers. (**3 Marks**)

4. A **deque** (pronounced deck) is a data structure for a double-ended queue. That is, it is a list to/from which we can make *push* and *pop* at/from either end.

   (a) Draw a picture of your deque, and illustrate its use for some example. This will help you to figure out its interfaces and implementation. (**2 Marks**)

   (b) Write the code for the deque interface. That is, show the struct and function definitions. (**2 Marks**)

   (c) Write the code for the deque implementation. That is, show how the functions defined above will work. (**2 Marks**)

   (d) Write a small client program to test whether your implementation works correctly. (**2 Marks**)

   (e) List 2 real-life applications in which it might be appropriate to use a deque. (**2 Marks**)

5. Suppose that you are part of a team developing a financial application. One of the functions used in the application is a calculator, for infix arithmetic expressions. For example, given the expression $5 * (((9 + 8) * (4 * 6)) + 7)$, the function outputs the result 2075. You are supposed to design the data structures and implementation for this calculator function. The input may be any arithmetic expression. Your function should read the expression from standard input, perform the computation and output the result.
   (HINT: Think about using 2 stacks, one for pushing the operands and the other for the operators. Whenever an operator or ')' is encountered in the input, you could pop the top 2 operands from the stack, evaluate their value and push the result back onto the stack. For example, when given the expression $(a + b)$, push '(' and 'a' onto the operand stack, push '+' onto the operator stack. Then push 'b' onto the operand stack. When ')' is encountered, pop 'b', 'a' and '(' from the operand stack, pop '+' from the operator stack, perform the addition and push the result back onto the operand stack. When no further input is encountered, return the top of the operand stack as the result.)

   (a) Draw a picture of your data structures, and illustrate their use for the above given expression (that evaluated to result 2075). This will help you to figure out the implementation. (**2 Marks**)

   (b) List your assumptions regarding the input, in order for your function to work correctly. (**2 Marks**).

   (c) Write the pseudo-code for your function. (Recall the discussion after the Quiz1 - Very-high level descriptions will be considered as comments, not as pseudo-code). (**4 Marks**).

   (d) Show the places in the pseudo-code, where you will perform checks to ensure that the input satisfies your assumptions. Also show how you will determine whether the given expressions are well-formed (valid) or not. Some examples of invalid expressions are: ((a+b), (a+b)), a+b*, etc. (**2 Marks**).

6. Attach your handwritten sheet (with Roll number clearly visible) to this answer paper. (**1 Mark**)

---

7. Take Home (Individual assignment): Find a computer to implement and test your solutions to Questions 4 and 5 above. Submit your code and test cases used, on Moodle, by 9 pm on Thursday (25th Feb 2010). Check the Moodle page for instructions from the TAs on naming conventions etc. (**10 Marks**)

   (NOTE:

   (a) You should implement the pseudo-code that you have written in the midsem answers, not your friend's! You may fix bugs in your psuedo-code logic to ensure that the implementation works correctly.

   (b) Sophisticated copy-detection software is being used. If A has copied from B, with or without B's knowledge, both A and B will get Zero for the entire Midsem. Absolutely no cribs or explanations will be entertained.

)
