

# CMPUT 115 Section A3 Final Exam

December 18, 2000  
Instructor: Jonathan Sillito

## Instructions:

- This is a closed book, no notes exam.
- Put all of your answers in the space provided.
- Be sure to write your student id number on each internal page.
- Please do not open the exam until you are instructed to do so.
- Good luck!



05152  
CMPUT 115 (A3)  
SILLITO, J.  
DEC 00 FINAL  
PAGES: 8

First Name:

Last Name:

1. [2 Marks] Draw the binary search tree that would result if the following characters were inserted: E, C, A, D, B, Z.

Question 2 refers to the following procedure.

```
procedure visitTree( node : BSTType );  
begin  
  if node <> nil then begin  
    visitTree( node^.right );  
    visitTree( node^.left );  
    writeln( node^.data );  
  end;  
end;
```

2. [2 Marks] Assume that the above procedure was called with the root of the tree you constructed in question 1, what would the output be?

Questions 3 and 4 refer to the following procedure. Note that  $N$  is a constant.

```

type
  dataArray = array[1..N] of integer;

procedure f( var A: dataArray );
var
  i,j,t:integer;
begin
  for i := 1 to N-1 do
    for j := i+1 to N do
      if A[i] < A[j] then begin
        t := A[j];
        A[j] := A[i];
        A[i] := t
      end;
    end;
  end;
end;

```

3. [2 Marks] Assume that the array  $A$  has length 6 (that is  $N = 6$ ) and the values in the array are:

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]
-4	5	8	-2	6	3

What will the array  $A$  look like after the call  $f(A)$  returns? Put your answer in the following diagram:

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]

4. [2 Marks] What is the maximum number of times the code inside the **if** statement (in the method  $f()$  above) could be executed? Express your answer in terms of  $N$ .

Questions 5 and 6 assume the following type definitions.

```
BSTElementType = string;  
BSTPointer = ^BSTNode;  
BSTNode = record  
  Data : BSTElementType;  
  Left, Right : BSTPointer  
end;
```

```
BSTType = BSTPointer;
```

5. [3 Marks] Write a **recursive** function that returns the size (that is, the number of nodes) in a tree.

```
function size( root : BSTType ): integer;
```

6. [3 Marks] Write a **recursive** function that returns the number of leaves in a tree.

```
function leaves( root : BSTType ): integer;
```

7. [3 Marks] Suppose you have a hash table implemented using an array ( $T$ ) of size 10, that stores integers using the hash function  $h(i) = i \bmod 10$ . Collisions are handled using chaining.

Use the diagram below to describe the state of the hash table after each of the integers 33, 43, 8, 128, 18 and 13 are added.

T[0]	
T[1]	
T[2]	
T[3]	
T[4]	
T[5]	
T[6]	
T[7]	
T[8]	
T[9]	

8. [3 Marks] Specify (using Pascal types) the type of linked structure you would use to implement a queue data structure?

type

Question 9 assumes the following definitions.

```
type
  { Hash table implementation using chaining. }
  HashElementType = record
    Key : string;
    Value : string
  end;

  HashNodePointer = ^HashNodeType;
  HashNodeType = record
    Data: HashElementType;
    Next: HashNodePointer;
  end;

  HashType = array [0..MAXSIZE-1] of HashNodePointer;

  function HashValue( Key:string ): integer;
```

9. [4 Marks] Implement the HashPut procedure for the Hashtable unit. Recall that the HashPut procedure is used to add a key-value pair to a hash table. The function HashValue returns a hash value for a string. You may use the HashValue function in your HashPut procedure.

```
procedure HashPut(var Table:HashType; Key:string; Value:string);
```

10. [2 Marks] An array A initially contains the values:

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
3	11	2	1	5	3	13	10	9

How will the array look after the `heapify` operation has been carried out? Put your answer in the following diagram:

A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]

Question 11 refers to the following recursive function. Recall that `mod` is the remainder operator, as an example: `12 mod 5` is 2.

```
function g( n : integer ): integer;
begin
  if n <= 1 then g := 3
  else if (n mod 2) = 0 then g := g( n div 2 ) + g( n-1 )
  else g := g( n-1 );
end;
```

11. [2 Marks] What value would be returned by the call `g(4)` to the function above?

12. [1 Mark] One way to sort a collection of values is to insert each of them into a binary search tree. Approximately, what is the *average* case time complexity of this sorting method? The following possible answers are expressed as functions of  $N$ , where  $N$  is the number of elements to be sorted.

- A.  $N^2$                       B.  $N(N-1)/2$                       C.  $N \log_2 N$                       D.  $N$

13. [1 Mark] If the sorting method described in question 12 was used on a collection of elements, what type of traversal of the tree would return the elements in sorted order?

- A. Inorder                      B. Postorder                      C. Preorder                      D. Levelorder

Student id: \_\_\_\_\_

*Extra paper for rough work*