UNIVERSITY OF DUBLIN

TRINITY COLLEGE

Faculty of Engineering and Systems Sciences

Department of Computer Science

B.A. (Mod) Computer Science **Trinity Term 2000**

SF Examination

2BA2 -- Programming Techniques

Saturday 27th May **Sports Hall** **14.00 - 17.00**

Dr. Hugh Gibbons

Attempt FOUR Questions.

(In presenting programs explain clearly the design of the Eiffel code)

Qs. 1.

a) Present an Eiffel routine

```
Search(a:ARRAY[G]; L,H:INTEGER; x:G) is
   require
      Ordered:  Is_Ordered(a,L,H)
   ensure
      Found_it: (found implies equal(A.item(index),x))
      Failed:   (not found implies
                   ((L-1 <= index and index <= H) and
                   (index = L-1 or else A.item(index) < x) and
                   (index = H or else x < A.item(index+1))))
```

that binary searches an array section, a[L..H], for an item, x.

b) Present an Eiffel routine,

`remove(a:ARRAY[G]; L,H:INTEGER; x:G)`

that will remove the last occurrence, if any, of an item, x, in an ordered array, a.

e.g. If the array contained the words,

adam, andrew, dave, dave, don, fred, john, oscar

then the procedure in removing the word, "dave", should remove the occurrence at index, 4, assuming the array in indexed from 1 to 8.

Suggestion:

Use an appropriate binary search routine that first finds the item x.

c) Present an Eiffel routine

`remove_all (a:ARRAY[G]; L,H:INTEGER; x:G)`

that removes all occurrences, if any, of an item, x, in an array, a.

Qs. 2.

a) Present an Eiffel routine

```
partition(L0, R0 : INTEGER; pivot: G)
  require
    -- A ≠ void
  ensure
    -- A[L0 .. R] ≤ pivot ≤ A[L .. R0]
    -- and perm(A[L0, R0], Ain[L0, R0] )
```

that will partition an array, A, into 2 sections such that the items in the left section are at most equal to the pivot item, pivot, and the items in the right section are at least equal to the pivot item, pivot. Also, after partition, the final array, A, is a permutation of the initial value of the array, A_{in} . Assume that the array, A, and the 'markers' L and R are attributes of the surrounding class.

Show how the routine would partition the array of characters,

P A R T I T I O N S

b) Using the routine, partition, present an Eiffel routine

qsort(left, right : INTEGER)

that will Quicksort the array section A[left, right] of the array attribute, A.

c) Using the routine, partition, present an Eiffel routine

find (k, left, right : INTEGER)

that finds the k^{th} smallest item in the array section A[left..right].

Qs. 3.

In the N-Queens problem, one has to place N queens on an NxN chessboard so that no queen can take another. Present an Eiffel program that will find a solution to the N-Queens problem with the extra constraint that no queen lies on any of the 2 main diagonals.

Qs. 4.

Assume we are given the classes, LIST_STRING and NODE with the following short forms

```
class interface LIST_STRING
create
    empty
feature
    empty
    -- make string empty
        ensure
            length = 0

    length : INTEGER

    is_empty : BOOLEAN

    prepend(ch : CHARACTER)
    -- put character, ch, at front

    item(k : INTEGER):CHARACTER
    -- returns the character at position, k
        require
            0 < k and k <= length

    join(ls : LIST_STRING)
    -- join the items of ls to end of Current

    copy(s : STRING)
    -- copy an Eiffel string to Current

    equal(ls : LIST_STRING):BOOLEAN
    -- is the string, ls, equal to Current,
    -- character by character

    substring(i,j : INTEGER):LIST_STRING
        require
            -- 0 < i ≤ j ≤ length
        -- if Current has characters
        -- "c1c2 .. cn" then
        -- substring(i,j) = "cici+1 .. cj"

end -- LIST_STRING
```

```
class interface NODE
feature
    item : CHARACTER

    next : NODE

    set_item(ch:CHARACTER)

    set_next(n : NODE)

end -- NODE
```

Implement the class, LIST_STRING, using linked nodes.

Use linked list diagrams to explain the routines.

Qs. 5.

Assume that a Directed Graph is stored as an adjacency list. Present Eiffel routines that will:

a) Read in the graph from a file

b) Depth First Traverse the Directed Graph

c) Show the output of the routine for Depth First Traverse, starting with vertex 1, when the Directed Graph is as follows:

```
{   (1,6), (1,5),
    (2,3), (2,7),
    (3,4), (3,5),
    (4,1), (4,2), (4,5), (4,7),
    (5,6),
    (7,5) }
```

Qs. 6.

Assume we have a class defining nodes of a binary search tree with interface

```
class interface BIN_NODE[G]
feature
   item : G
   left, right : BIN_NODE[G]
   build(v: G; L, R : BIN_NODE[G])
end
```

a) Present an Eiffel function,
   `delete(x:G; bt : BIN_NODE[G])`
   that will delete an item, x, from a binary search tree with root, bt.
   Explain the algorithm with the use of tree diagrams.

b) Present a non-recursive Eiffel routine,
   `nr_inorder(bt : BIN_NODE[G])`
   that will inorder a binary search tree with root, bt.

© UNIVERSITY OF DUBLIN 2000