University of Glasgow

Wednesday 28 April 2021
Available from 09:15 BST
Expected Duration: 1 hour 30 minutes
Time Allowed: 3 hours
Timed exam within 24 hours

DEGREES OF MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

ALGORITHMS AND DATA STRUCTURES 2
COMPSCI2007

Answer all 5 questions

This examination paper is an open book, online assessment and is worth a total of 60 marks

1. Algorithm **F** takes as input an array of integers A and two indices p, r of A. It is described by the following pseudocode:

```
1: F(A, p, r)
2:   if p < r
3:     x := A[p]
4:     A[p] := A[r]
5:     A[r] := x
6:     F(A, p+1, r-1)
```

(a) Briefly explain what algorithm **F** implements. [5]

Solution: This algorithm reverses the elements in the subarray $A[p..r]$.

(b) What is the output of $\mathbf{F}(A,0,6)$, where $A = [1,5,7,9,3,4,1]$? [2]

Solution: The output is array $[1,4,3,9,7,5,1]$.

(c) Draw the recursion trace for $\mathbf{F}(A,0,6)$. [3]

Solution:

$\mathbf{F}(A,0,6)$	$A = [1,4,3,9,7,5,1]$
$\mathbf{F}(A,1,5)$	$A = [1,5,3,9,7,4,1]$
$\mathbf{F}(A,2,4)$	$A = [1,5,7,9,3,4,1]$
$\mathbf{F}(A,3,3)$	

(d) Is **F** tail recursive? Justify your answer. [2]

Solution: Yes, as the recursive call is the last operations performed by the algorithm.

(e) Is **F** an in-place algorithm? Justify your answer. [2]

Solution: Yes, as the subarray of the input array A is reversed using no auxiliary data structure.

(f) Write an iterative version of algorithm **F** and compute its complexity. Suppose n is the size of the input array. Use big-Oh notation. [6]

Solution:

```
1: F-ITER(A, p, r)
2:   while p < r
```

```
3:      x := A[p]
4:      A[p] := A[r]
5:      A[r] := x
6:      p := p + 1
7:      r := r − 1
```

The running time is $O(n)$.

2. For each of the following statements, prove whether it is true or false.

(a) $29 = O(\log_2 n)$ [3]

Solution: True. To prove $29 \leq c \log_2 n$ for $n \geq n_0$, pick $c = 10$ and $n_0 = 2^3 = 8$.

(b) $\max(n^3, 10n^2) = O(n^2)$ [3]

Solution: False. Suppose that there were constants n_0 and c such that for all $n \geq n_0$, $\max(n^3, 10n^2) \leq cn^2$. If $n \geq 10$, we have $\max(n^3, 10n^2) = n^3 \leq cn^2$. Then $c \geq n$. But since c is a constant, the inequality does not hold for any n.

3. **(a)** Briefly describe using your own words the counting sort algorithm (expected word count: 100). Illustrate your description by sorting the list: 3, 7, 4, 2, 1, 5, 4. [6]

Solution: The counting sort algorithm is a sorting algorithm where the inputs are arrays of integers in the range 0 to $k \geq 0$. It consists of three steps

1. Initialise a new array C of size $k+1$ and use it to count the occurrences of each element in the input.
2. Modify the counts to have a running sum. The modified count array indicates the position of each object in the output sequence.
3. Use the running sum to copy elements to an auxiliary array B containing the sorted sequence.

The steps performed to sort input $A = [3,7,4,2,1,5,4]$ are the following:

1. Counts are stored in C by the first for loop: $C = [0,1,1,1,2,1,0,1]$.
2. The running sum is computed by the second for loop: $C = [0,1,2,3,5,6,6,7]$.
3. Values from A, starting from $A[6]$, are copied in order in B and C is updated as follows:

- $B = [*,*,*,*,4,*,*]$ and $C = [0,1,2,3,4,6,6,7]$,
- $B = [*,*,*,*,4,5,*]$ and $C = [0,1,2,3,4,5,6,7]$,
- $B = [1,*,*,*,4,5,*]$ and $C = [0,0,2,3,4,6,6,7]$,
- $B = [1,2,*,*,4,5,*]$ and $C = [0,0,1,3,4,6,6,7]$,
- $B = [1,2,*,4,4,5,*]$ and $C = [0,0,1,3,3,6,6,7]$,
- $B = [1,2,*,4,4,5,7]$ and $C = [0,0,1,3,3,6,6,6]$,
- $B = [1,2,3,4,4,5,7]$ and $C = [0,0,1,2,3,6,6,6]$.

(b) Briefly describe using your own words the radix sort algorithm (expected word count: 50). Illustrate your description by sorting the list: 802, 256, 958, 938, 693, 405, 684, 854. [6]

Solution: The radix sort algorithm is a sorting algorithm where the inputs are assumed to have d digits. It iteratively sorts on each digit using a stable sorting algorithm (typically counting sort), starting from the least-significant digit (left to right).

The steps performed to sort input $A = [802,256,958,938,693,405,684,854]$ are the following:

- Stable sort on third digit to get $A = [80\mathbf{2},69\mathbf{3},68\mathbf{4},85\mathbf{4},40\mathbf{5},25\mathbf{6},95\mathbf{8},93\mathbf{8}]$.
- Stable sort on second digit to get $A = [8\mathbf{0}2,4\mathbf{0}5,9\mathbf{3}8,8\mathbf{5}4,2\mathbf{5}6,9\mathbf{5}8,6\mathbf{8}4,6\mathbf{9}3]$.
- Stable sort on first digit to get $A = [\mathbf{2}56,\mathbf{4}05,\mathbf{6}84,\mathbf{6}93,\mathbf{8}02,\mathbf{8}54,\mathbf{9}38,\mathbf{9}58]$.

4. Illustrate the result of each operation in the sequence PUSH(S, 5), PUSH(S, 2), PUSH(S, 4), POP(S), PUSH(S, 7), and POP(S) on an initially empty stack S. Consider the following two cases:

(a) Stack S is implemented by array $S[0..5]$. [3]

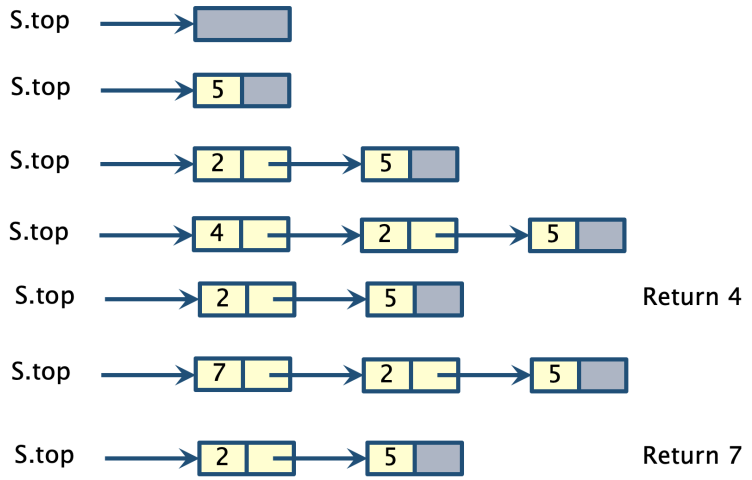
Solution:

- S is empty, S.top = -1, $[*,*,*,*,*,*]$;
- PUSH(S, 5), S.top = 0, $[5,*,*,*,*,*]$;
- PUSH(S, 2), S.top = 1, $[5,2,*,*,*,*]$;
- PUSH(S, 4), S.top = 2, $[5,2,4,*,*,*]$;
- POP(S), return 4, S.top = 1, $[5,2,*,*,*,*]$;
- PUSH(S, 7), S.top = 2, $[5,2,7,*,*,*]$;

- POP(S), return 7, S.top = 1, $[5, 2, *, *, *, *]$;

(b) Stack S is implemented by a linked list. [3]

Solution:



5. **(a)** Briefly explain with your own words what a hash table is, its main components and its operations? Expected word count: 100. [3]

Solution: A hash table is a data structure for the storage of key-value pairs. It consists of two main components:

- an array $T[0,..,m-1]$ of fixed size and
- a hash function $h : U \to \{0, 1, \ldots, m-1\}$ mapping keys from a universe U to slots in T, where $m \ll |U|$.

Hash tables support **INSERT**, **DELETE**, and **SEARCH** operations in $O(1)$ time on average.

(b) Briefly explain with your own words what a *hash collision* is and give an example application for which hash table storage would be suitable. Expected word count: 50. [3]

Solution: When two keys are mapped to the same position in array T, we have a *hash collision*. Ideally, the hash function is easy to compute and no collisions occur. A hash table is an effective data structure for implementing dictionaries, database indexes, and caches.

(c) Briefly describe with your own words the chaining method of hash collision resolution

and give one advantage of this method. Expected word count: 70. [2]

Solution: In collision resolution by chaining, each hash-table slot $T[j]$ contains a linked list of all the keys whose hash value is j. The linked list can be either singly or doubly linked. Some advantages of this collision resolution method are:

1. simple to implement;
2. hash table never fills up, we can always add more elements to the chain;
3. less sensitive to the hash function or load factors.

(d) Show the resulting hash table if keys 5, 28, 19, 15, 20, 33, 12, 17, 10 are inserted into an initially empty hash table assuming collisions are resolved by chaining. Suppose the table has 9 slots, and let the hash function be $h(k) = k \mod 9$. [3]

Solution: The hashes of the keys and the resulting hash table T after the sequence of **INSERT** operations are computed as follows:

k	$h(k)$
5	5
28	1
19	1
15	6
20	2
33	6
12	3
17	8
10	1

T	
0	
1	$\rightarrow$ 10, 19, 28
2	$\rightarrow$ 20
3	$\rightarrow$ 12
4	
5	$\rightarrow$ 5
6	$\rightarrow$ 33, 15
7	
8	$\rightarrow$ 17

(e) Briefly describe with your own words the open addressing method of hash collision resolution. Expected word count: 70. [2]

Solution: In open addressing, all elements occupy the hash table itself. That is, each element of table T contains either a key or NIL. When searching for an element, we systematically examine table slots until either we find the desired element or we can confirm that the element is not in the table.

(f) Show the resulting hash table if keys 50, 700, 76, 85, 92, 73, 101 are inserted into an initially empty hash table assuming collisions are resolved by open addressing with *linear probing*. Suppose the table has 7 slots, and let the hash function be $h(k) = k \mod 7$. [3]

Solution: The hashes of the keys and the resulting hash table T after the sequence of **INSERT** operations are computed as follows:

k	$h(k)$
50	1
700	0
76	6
85	1
92	1
73	3
101	3

0	700
1	50
2	85
3	92
4	73
5	101
6	76

END OF QUESTION PAPER