The University of Nottingham
Malaysia

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, FULL YEAR 2022-23

ALGORITHMS CORRECTNESS AND EFFICIENCY

Time allowed TWO hours

Candidates must NOT start writing their answers until told to do so

Answer ALL questions

No calculators are permitted in this examination.

Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject specific translation dictionaries are not permitted.

No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.

DO NOT turn examination paper over until instructed to do so

INFORMATION FOR INVIGILATORS: Collect both the exam papers and the answer booklets at the end of exam.

1. Topic: Amortized Analysis

[Total: 25 marks]

Let us consider a list of numbers implemented as a stack with the following operations:

Operation	Description	Cost
Add(*a*)	Appends the number *a* to the front of the list (push operation on the stack)	1
Aggregate()	Removes all numbers from the list, computes their average, and then adds the average back onto the list	The cost is equal to the number of integers in the list plus one
Pop()	Pop is used by the aggregate operation to remove one element from the list	1

a) What amortized analysis is used for? How is it different from the asymptotic analysis?
[3 marks]

b) Assuming we perform *n* operations on the list, how **large** might the running time of one of the operations be? Justify your answer.
[4 marks]

c) Show that the amortized cost of these operations is small using the accounting method.

i First, give the amount that you will charge Add() and the amount that you will charge Aggregate().
ii Next show how you will use these charges to pay for the actual costs of these operations.
iii Finally, write down the amortized cost per operation.

[9 marks]

continue on next page

d) Use the potential method to find the amortized cost per operation. Let L_i be the list after the i-th operation and let $num(L_i)$ be the number of items on L_i. Use the following potential function:

$$\varphi_i = num(L_i)$$

i. First, show that this potential function is valid (i.e. $\varphi_0 = 0$ and $\varphi_i \geq 0$ for all i)

ii. Next use this potential function to calculate the amortized costs of Add and Aggregate (Recall that $a_i = c_i + \varphi_i - \varphi_{i-1}$, where a_i, is the amortized cost of the i-th operation and c_i is the actual cost)

[9 marks]

2. Topic: Recurrence Relations

[Total: 25 marks]

a) Consider the following function:

```
TREE-SEARCH(x, k)
1  if x == NIL or k == x.key
2       return x
3  if k < x.key
4       return TREE-SEARCH(x.left, k)
5  else return TREE-SEARCH(x.right, k)
```

Write the recurrence relation for TREE-SEARCH. [3 marks]

b) Give an asymptotic solution for the following recurrence:

$T(n) = 4T\left(\frac{n}{3}\right) + n^2, with\ T(1) = 1$ using the tree method.

i. Draw the first 3 levels of the tree.

[3 marks]

ii. What is the input size at level 6?

[2 marks]

iii. What is the complexity at level 5?

[2 marks]

iv. Find the Total complexity of the tree.

[5 marks]

c) Consider the recurrence $T(n) = 3T\left(\frac{n}{3}\right) + n, with\ T(1) = 1$

Use the substitution method to solve this recurrence and find the asymptotic upper bound. Do not forget to write down precisely what you are trying to prove and to include the base case, inductive hypothesis, and inductive step.

[10 marks]

3. Topic: String Matching

[Total: 25 marks]

a) Using the naïve string matching algorithm, how many character comparisons does NaiveStringMatch perform until all occurrences of pattern *P* are found in *T*?

- T = kbkbkbkbmkbkbm
- P = kbkbm

[5 marks]

b) Given the Pattern P = kbkbm, build an FSA that recognizes all occurrences of P in a given string. [5 marks]

c) Provide a pattern and a string that represent the **best-case** scenario for the KMP algorithm, (i.e., the algorithm runs in the fastest possible way for such a problem). [5 marks]

d) Provide a pattern and a string that represent the **worst-case** scenario for the KMP algorithm, (i.e., the algorithm runs in the slowest possible way for such a problem). [5 marks]

e) How many spurious hits does the Rabin-Karp string matching algorithm encounter in the text T = "3141512653849792" when looking for all occurrences of the pattern P = "26", working modulo q = 11 and over the alphabet $\Sigma = \{0, 1, 2,..., 9\}$? [5 marks]

4. Topic: Hash Table, Graph

[Total: 25 marks]

a) Suppose you are implementing a hash table and are trying to choose between using a probing strategy and using separate chaining (each slot is actually a linked list that can hold as many records as needed). You will use the same hash function and hash table size no matter which strategy you select.

i. A *primary collision* occurs when two records map to the same home slot in the table. Will the number of primary collisions be lower if you choose probing or if you choose separate chaining? Justify your conclusion carefully.

[3 marks]

ii. The drawback of using separate chaining is that you will have to perform a linear traversal of the linked list to find a record, and linear traversals are slow. Assuming that a small but significant number of primary collisions will occur, would searching be more efficient if you chose **linear probing** instead of separate chaining? Justify your conclusion carefully.

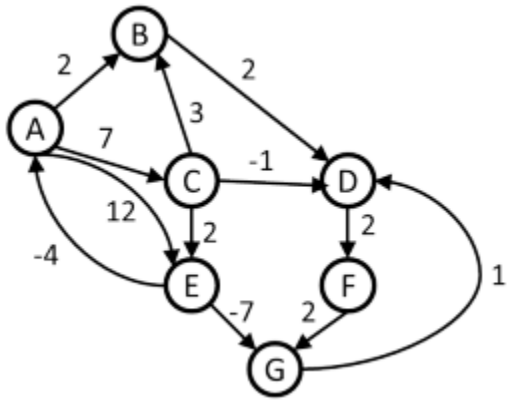
[4 marks]

iii. Reconsider part (ii) under the assumption that you would choose **quadratic probing** as an alternative to separate chaining. Justify your conclusion carefully.

[4 marks]

continue on next page

b) Consider the following directed, weighted graph with negative weight edges:



i Step through **Dijkstra's algorithm** to calculate *supposedly* shortest paths from A to every other vertex. Show your steps in the table below. Cross out old values and write in new ones, from left to right within the cell, as the algorithm proceeds. List the vertices in the order which you marked them known.

[6 marks]

Known vertices (in order marked known): **A**, _, _, _, _, _, _

Vertext	Distance	Path
A	**0**	
B	**2**	**A**
C	**7**	**A**
D	**∞**	
E	**12**	**A**
F	**∞**	
G	**∞**	

ii **Dijkstra's algorithm** found the wrong path to some of the vertices. For just the vertices where the wrong path was computed, indicate **both** the path that was computed and the correct path.

[6 marks]

iii What *single* edge could be removed from the above graph such that **Dijkstra's algorithm** will compute the correct answers for all vertices in the remaining graph.

[2 marks]