Algorithms and their Applications CS2004 (2020-2021)

Dr Mahir Arzoky

18.1 Exam Revision

Introduction

- ❑ This lecture is a revision lecture for the examination for this module
- ❑ The exam will be timed, online and open book
- ❑ The exam will be on WiseFlow
- ❑ Your exam timetable can be found here: https://students.brunel.ac.uk/study/exam-dates
- ❑ Refer to BlackBoard CS2004 Homepage for more information
- ❑ This lecture covers:
  - ❑ The assessment
  - ❑ The format of the exam
  - ❑ Exam topics
  - ❑ Brief revision of some topics
  - ❑ Some example questions

Assessment

- The overall structure for the assessment is as follows:
  - 60% Coursework
    - Task #1 (Class tests)
    - Task #2 (CodeRunner Examination)
  - 40% Exam (pending)
  - Pass both elements (coursework and exam) at E- (30%) grade
  - Pass the module at D- (40%) grade

Exam

- ❑ A three hour exam
  - ❑ Using **WiseFlow**
- ❑ This will consist of:
  - ❑ 5 essay-type questions
- ❑ All questions should be attempted

What to Revise – Part 1

- The exam will cover the theoretical aspects of the module
- There will be no programming needed
    - No questions on Java or Eclipse
- However you *may* need to understand and/or write pseudo code

What to Revise – Part 2

- ❑ Topics could include (not restricted to):
  - ❑ Algorithmic concepts
    - ❑ What is an algorithm, a program, etc…
  - ❑ Time Complexity and Asymptotic Notation
    - ❑ $T(n)$ and $O(n)$
  - ❑ Data structures
    - ❑ Stacks, lists, arrays, queues, etc…
  - ❑ Sorting Algorithms
    - ❑ Bubblesort, Quicksort, etc…

What to Revise – Part 3

- Topics could include (not restricted to):
  - Graph Traversal Algorithms
    - Depth First, Breadth First, A*, MST, etc…
  - Search
    - Search, Search Space, Fitness, Parameter optimisation, etc...
  - Heuristic Search Methods
    - HC, SHC, RRHC, SA, ILS, etc...
  - Evolutionary Computation and Other Methods
    - Genetic Algorithms, PSO, ACO, etc...
  - Applications
    - TSP, Bin Packing, Data Clustering, etc...

What to Revise – Part 4

- Past Papers
  - Several years past papers are available
  - Very useful to revise using them
  - I cannot provide the answers due to University policy!
  - However I am happy to discuss any answers with you

Exam – Part 1

- ❑ Essay type questions
    - ❑ 5 questions of 20 marks
    - ❑ Spend **approximately** 36 minutes per question

Exam – Part B – Part 2

- Good strategies for essay type questions:
  - Read through all of the questions first
  - Answer the ones that you know first
  - Do **NOT** spend too much time on a single question
  - Sketching a draft answer can help in laying out complex answers
  - Delete anything you do not want marked

Further Notes of Exams…

- ❑ Start the test on time
- ❑ Plan your time
- ❑ Try to concentrate and ignore what is going on around you…
- ❑ Invigilation during the examination
- ❑ Feel free to ask questions

Requested Topics

- ❑ Computational Complexity and Asymptotic Analysis
- ❑ Counting primitive operations
- ❑ Big-T and Big-O
- ❑ Drawing the clustering arrangement of a given cluster representation

Asymptotic Algorithm Analysis

- Experimental studies can have limitations
- Asymptotic Analysis uses a high-level description of the algorithm instead of an implementation
- The performance of an algorithm is evaluated in terms of input size
- We calculate, how does the time taken by an algorithm increases with the input size
- To perform asymptotic analysis
  - We find the worst-case number of primitive operations executed as a function of the input size, $\mathrm{T}(n)$
  - We express this function with Big-Oh notation

T(n) and O(n)

❑ We estimate the running time/computation of an algorithm

❑ We refer to this resultant formulae as T(*n*) where n is the size of the input

❑ We measure T(*n*) in terms of primitive operations

❑ We can use *T*(*n*) to compute a very important property called the Big-O (O(*n*))

❑ Big-O notation defines an upper bound of an algorithm (worst-case)

❑ From T(*n*) to Big-O - all constants and lower order terms are dropped

Primitive Operations – Part 1

- ❑ Basic computations performed by an algorithm
- ❑ Largely independent from the programming language
- ❑ Examples:
  - ❑ Evaluating an expression (x>y?)
  - ❑ Assigning a value to a variable (x=0)
  - ❑ Indexing into an array (for A[0] or A[i] we might use the mathematical notation a_0 or a_i)
  - ❑ Calling a method
  - ❑ Returning from a method

Primitive Operations – Part 2

- ❑ Consider the following lines of Pseudo-Code:

```
Let a be an array of size n where $a_i$=0
```

n for creating the array and n for setting to zero ***T*(*n*)**= 2n

```
Let x = 10
```

One operation (set) ***T*(*n*)= 1**

```
Let y = x
```

One read and one write ***T*(*n*)**= 2

```
Let z = x + y
```

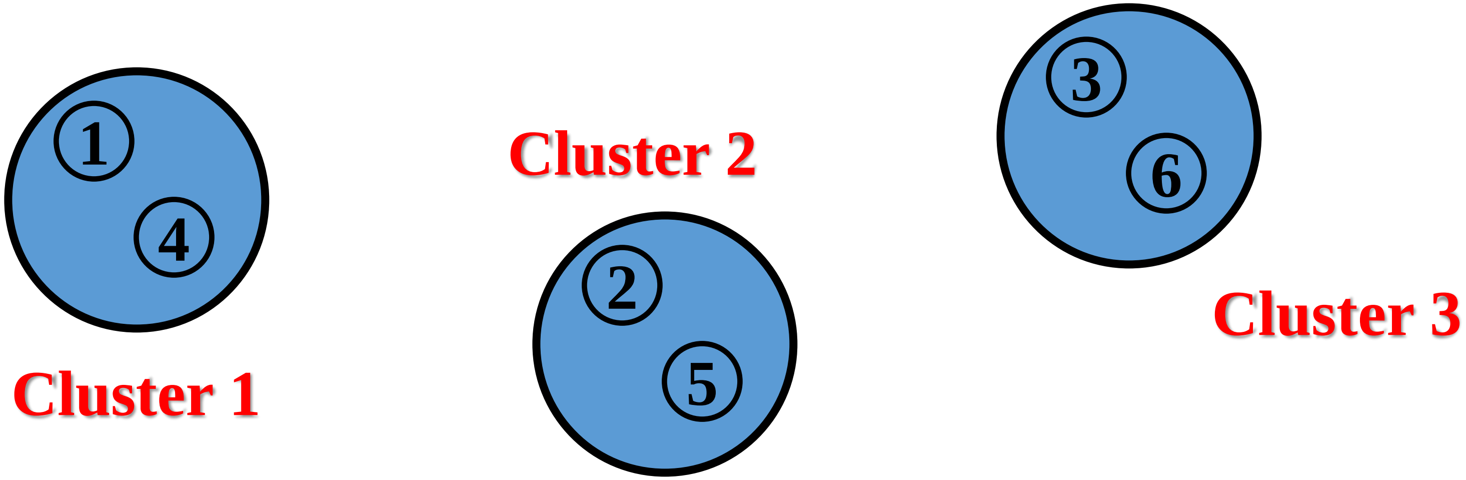
Two reads, one arithmetic and one write/set ***T*(*n*)= 4**

```
For i = 1 to n   (n operations)
    Let $a_i$ = $a_i$ + x + y + 10   (=9 – repeated n times for For loop T(n)= 9n)
                                     (1 [2] for the write, 5 [4] reads, 3 operators)
End For
```

No count – indicates the end of the loop

Representing a Cluster

- ❑ A cluster will be represented as a vector C where $c_i=j$ means that object/item/row i is in cluster j
- ❑ For example C = {1,2,3,1,2,3} ($k=3$)



Sample Questions and Answers

Example Essay Type – Part 1

- ❑ Consider the following algorithm:

```
Algorithm 1. MaxArray(A)
Input: An n row by m column Array A
1) Let max = element 1,1 (A(1,1)) of Array A
2) For i = 1 to n
3)    For j = 1 to m
4)       If A(i,j) > max Then
5)          Let max = A(i,j)
6)       End If
7)    End For
8) End For
Output: max- the largest element in array A
```

Example Essay Type – Part 2

- ❑ For a total of 15 marks:
  - ❑ Describe the algorithm in words (not pseudo code)
  - ❑ Compute $T(n)$ for algorithm 1
  - ❑ Compute $\mathrm{O}(n)$ for algorithm 1
  - ❑ How would you modify the algorithm to create a new algorithm `MinArray`?

Example Essay Type – Answers

- ❑ Describe the algorithm in words (not pseudo code)
- ❑ The algorithm is designed to locate the largest value in an array, passed as a parameter. It assumes that the maximum is equal to the first element (1,1) and then systematically examines each element in turn, updating the maximum if the current item under scrutiny is larger than the maximum. This maximum value is then returned by the algorithm.

Example Essay Type – Answers

- ❑ For T(n) count all variables reads, writes and operators, note we have two input sizes n and m

```
Algorithm 1. MaxArray(A)
Input: An n row by m column Array A
Let max = element 1,1 (A(1,1)) of Array A → 2
For i = 1 to n → n
    For j = 1 to m → n × (m)
        If A(i,j) > max Then → n × m × (5) [Assume worse]
            Let max = A(i,j) → n × m × (4)
        End If → none
    End For → none
End For → none
Output: max- the largest element in array A → none
```

Example Essay Type – Answers

- ❑ Compute $T(n)$ for algorithm 1
- ❑ For $T(n)$:
  - ❑ 2
  - ❑ n
  - ❑ nm
  - ❑ $5nm$
  - ❑ $4nm$
  - ❑ $T(n) = 10nm + n + 2 = n(10m+1) + 2$
- ❑ Compute $O(n)$ for algorithm 1
  - ❑ $O(n) = nm$
- ❑ How would you modify the algorithm to create a new algorithm `MinArray`?
- ❑ To create algorithm `MinArray`
  - ❑ We change the > on line 4 to a <
  - ❑ We would also rename the algorithm name and results variable

Another Example Essay Type

- Describe the main similarities and differences between a Genetic Algorithm (GA) and an Evolutionary Program (EP). [4 marks]

Example Essay Type – Answers

- Both maintain a population
- Only a GA has crossover
- Both have mutation but an EP has a much more complex mutation operator
    - All individual mutate in an EP
    - Only some in a GA
- Both have selection
    - However a GA usually uses the **Roulette Wheel** which allows an individual to be selected zero or more times
    - An EP uses **Tournament Selection** which allows an individual to be selected zero times or once

Lastly…

- ❑ Hopefully see you next year
- ❑ Good luck!
- ❑ For any questions do not hesitate to contact me!