# The University of Nottingham Ningbo China

Centre for English Language Education

Semester One, 2022-2023

## INTRODUCTION TO ALGORITHMS

Time allowed: 24 hours take-home exam

---

**This paper contains FOUR questions which carry equal marks.**

**Attempt ANY THREE questions.**

**Marks will be given for the best THREE answers.**

**Show all necessary steps in your solutions.**

*An indication is given of the weighting of each subsection of a question by means of a figure enclosed by square brackets, eg.* [5]*, immediately following that subsection.*

**No calculators are permitted in this exam**.

**ADDITIONAL MATERIAL:** *None.*

**INSTRUCTIONS:**

1. *Please typeset your answers neatly using a text editor (e.g.MS Word).* **Hand-written submissions will not be accepted**.
2. *Create a single PDF file of all pages of your answers to exam questions.*
3. *Name your file as: Your Student ID number_N086Final. For example: 20519999_N086Final.*
4. *Please complete the* **Academic Integrity Declaration**.
5. *Please upload the PDF file to Moodle drop-box (check email from Module Convenor for the link to drop-box).*
6. *Submission deadline:* **by 9.30 am on Friday 6 January 2023**.
7. *This work must be completed on your own. Plagiarism and collusion are regarded as very serious academic offences and will be treated as such.*

1 Suppose you have a machine that can only perform operation for addition.

(a) Write a recursive algorithm called `addition(x,n)` that takes two positive integers $x$ and $n$, and returns the value of $x \times n$ by adding $x$ to itself $n$ times. [5]

(b) Write a recursive algorithm called `power(x,n)` that takes two positive integers $x$ and $n$, and returns the value of $x^n$.

You must call the algorithm `addition(x,n)` in (a) as a sub-algorithm. [5]

(c) Trace your algorithms for power(3,4). You can use the sub-algorithm for computation directly and only show the process in the main algorithm. [5]

2 (a) Consider the list `L = [21, 8, 15, 39, 27, 3, 2, 36]`.

($i$) Show the process of sorting `L` in ascending order using merge sort.

($ii$) State the time complexity of merge sort.

($iii$) Which sorting scheme also has the same time complexity as merge sort? [5]

(b) ($i$) Show the process of sorting `L` in ascending order using insertion sort.

($ii$) Give one example of the best case for insertion sort using a list with same elements as in `L`. [5]

(c) Write a recursive algorithm called `maxDiff(L)`, that takes a sorted list $L$ with `length`$(L) \geq 2$ in acending order, and returns the maximum difference between two successive elements in the list.

For example, `maxDiff([2,3,8,15,21,27,36,39]) = 9`.

You can call algorithms `length`$(L)$ to find the length of the list, and `max(x,y)` to find the larger value out of two. [5]

3 Consider the binary tree T in Figure 3.1.

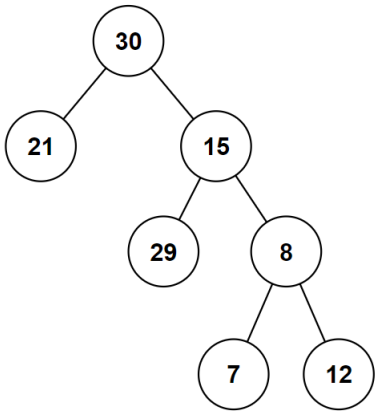


Figure 3.1

(a) $(i)$ Write the leaf and node notation for T.

$(ii)$ Write the list of all values in T using postorder traversal scheme.

$(iii)$ Write the list of all values in T using breadth first traversal scheme. [5]

(b) Write a recursive algorithm called `level(x,T)` that takes a value $x$ and a binary tree T, and returns the level to which the value $x$ belongs. Note that the tree levels begin from $0$. If $x$ does not occur in the binary tree, your algorithm should return $-1$.

You can call algorithm `search(x,T)` to search for $x$ in the tree first.

You can also call algorithm `max(x,y)` to find the larger of two values. [5]

(c) Show the process of reconstructing the binary tree T into a binary search tree with minimal depth. [5]

4 (a) Consider the graph in Figure 4.1.

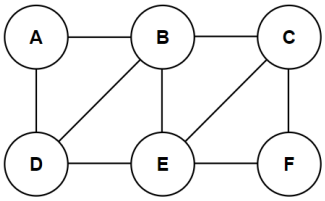


Figure 4.1

($i$) Determine whether it has Euler path, Euler circuit or neither.

($ii$) Is it a bipartite graph? Explain the reason.

($iii$) Draw any two spanning trees of this graph. [5]

(b) Consider the directed graph in Figure 4.2.

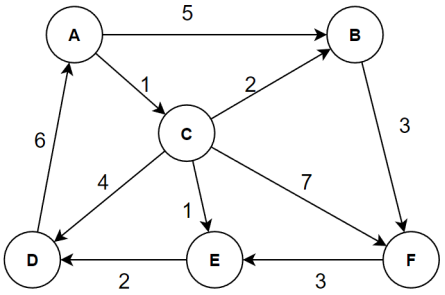


Figure 4.2

Apply Dijkstra's algorithm to find the shortest paths and minimum costs from vertex A to all other vertices. You need to clearly show the process. [5]

(c) Consider the undirected graph in Figure 4.3.

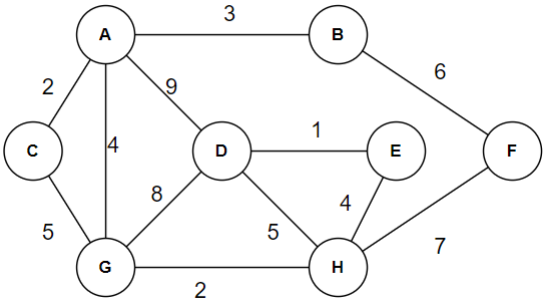


Figure 4.3

Apply Prim's algorithm (starting at Vertex A) to find the minimum spanning tree and minimum cost. You need to clearly show the process. [5]