Family Name: ..... **Other Names:** .....

Student ID: ..... **Signature** .....

COMP 261 : Test 2

28 May 2026 ** WITH SOLUTIONS **

Instructions

- Time allowed: **120 minutes**
- Attempt **all** the questions. There are 100 marks in total.
- Write your answers in this test paper and hand in all sheets.
- If you think a question is unclear, ask for clarification.
- This test contributes 30% of your final grade.
- You may use dictionaries and calculators.
- You may write notes and working on this paper, but make sure your answers are clear.

Questions	Marks	
1. Huffman Coding	[13]	
2. Lempel Ziv Coding	[8]	
3. Arithmetic Coding	[9]	
4. String Search	[17]	
5. Fast Fourier Transform	[8]	
6. Adjacency Matrix and Adjacency List	[15]	
7. Shortest Paths	[30]	
	TOTAL:	

Question 1. Huffman Coding **[13 marks]**

(a) **[9 marks]** The following text is to be encoded using Huffman coding, based on the letter frequencies in the text itself:

`banana_band_ana`

- Construct the relevant binary tree,
- Show the code it assigns to each character.

Note 1: If the two nodes have the same frequency in the priority queue, pick the node with the smaller character alphabetically. Specifically, "_" comes **before all the lowercase letters. For a parent node, using its left most leaf node to decide its alphabetial order if needed*

Note 2: When building a parent node, use the child node with the smaller frequency as the **left child.*

**Note 3: You can use a "+" or a circle to represent the non-leaf nodes when drawing a Huffman tree.*

Draw your Huffman tree below:

Frequencies: a:6, n:4, b:2, _:2, d:1 (Total: 15)
1. Combine {d(1), _(2)} → P1(3).
2. Combine {b(2), P1(3)} → P2(5).
3. Combine {n(4), a(6)} → P3(10).
4. Combine {P2(5), P3(10)} → Root(15).
(Exact tree shape depends on tie-breaking, but codes should be equivalent in length)

The code for each character is:
a: 11
n: 10
b: 00
_: 010
d: 011

(b) **[2 marks]** Encode the string using your codebook:

00 11 10 11 10 11 010 00 11 10 011 010 11 10 11

(c) **[2 marks]** In a Huffman codebook, no code can be a prefix of another code (e.g., if 'A' is 01, 'B' cannot be 011). Explain why this property is essential for the decoding process.

Without it, the decoder wouldn't know where one character ends and the next begins (ambiguity).

Question 2. Lempel-Ziv Coding **[8 marks]**

(a) **[3 marks]** Suppose that the following tuple sequence is an encoding result by Lempel-Ziv:
`[0, 0, 'w'][0, 0, 'o'][0, 0, 'd'][2, 2, 'l'][0, 0, 'y'][7, 3, 's']`

Which string is the correct decoding result? [ ] C
(A) woodlywoods (B) woolywoods (C) wododlywods (D) woodly_woods

(b) **[3 marks]** For the string: `ababa_abacaba`
Which option shows the correct Lempel-Ziv encoding? [ ] D

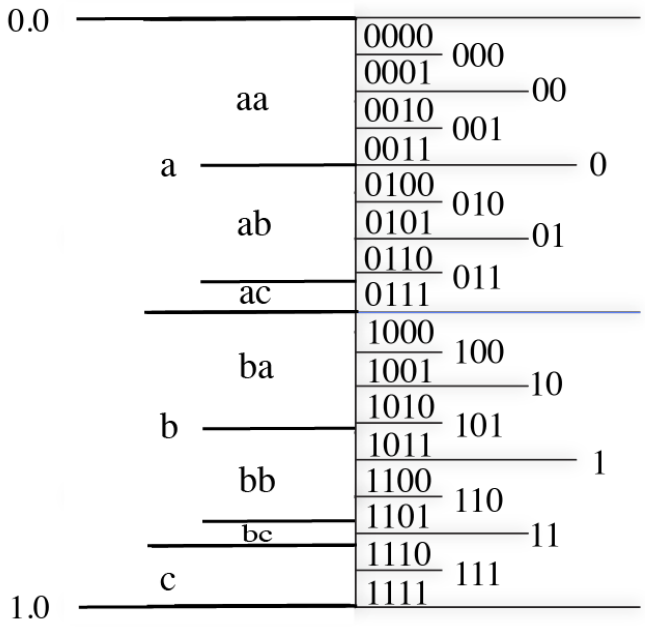
(A) `[0,0,'a'][0,0,'b'][2,2,'a'][0,0,'_'][5,4,'c'][2,2,'a']`
(B) `[0,0,'a'][0,0,'b'][2,3,'_'][5,3,'a'][0,0,'c'][3,2,'a']`
(C) `[0,0,'a'][0,0,'b'][2,2,'_'][1,1,'a'][5,3,'c'][3,2,'a']`
(D) `[0,0,'a'][0,0,'b'][2,2,'a'][0,0,'_'][6,3,'c'][10,3,'']`

(c) **[2 marks]** In Lempel-Ziv, we use a Search Window (the past data) and a Lookahead Buffer (the data to be encoded). If you increase the Search Window from 1,024 bytes to 1,024,000 bytes, how does this affect the compression ratio?

Ratio improves (more likely to find matches)

Question 3. Arithmetic Coding **[9 marks]**

Consider the alphabet of {a, b, c}, with probabilities {0.5, 0.4, 0.1}.



(a) **[3 marks]** For the string "aa", the final interval is [0.0, 0.25). What is the binary code to send?

[ ] B

(A) 0 (B) 00 (C) 01 (D) 001

(b) **[3 marks]** To transmit "c" (interval [0.9, 1.0)). What is the first bit to transmit? [ ] B

(A) 11 (B) 1 (C) 0 (D) nothing

(c) **[3 marks]** Huffman coding is known to be "optimal" for symbol-by-symbol encoding, yet Arithmetic coding often achieves better compression ratios. How does Arithmetic coding achieve this?

Arithmetic treats the whole message as one number, allowing a symbol to effectively take up a "fractional" bit (or something similar).

Question 4. String Search **[17 marks]**

(a) **[7 marks]** Show the partial match table **M** generated for the KMP algorithm for the following search string **S**:

The string S: ″aabaaba″

Index	0	1	2	3	4	5	6
S	a	a	b	a	a	b	a
M	-1	0	1	0	1	2	3

(b) **[10 marks]** Show the step-by-step process of the KMP algorithm using your partial match table. Search through the text **T** until a full match is found.

Pos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
T	a	a	b	a	a	b	**c**	a	a	b	a	a	b	a	a	b	a	x	y

Match Attempt 1:
Start Position in T: **T[0]**
Matched Part: ″aabaab″
Return a value? No

Match Attempt 2:
Start Position in T: T[3]
Matched Part: ″aab″
Return a value? No

Match Attempt 3:
Start Position in T: T[6]
Matched Part: None
Return a value? No

Match Attempt 4:
Start Position in T: T[7]
Matched Part: ″aabaaba″
Return a value? Yes: 7

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 5. Fast Fourier Transform **[8 marks]**

(a) **[4 marks]** When using the Fast Fourier Transform (FFT) to multiply two polynomials $A(x)$ and $B(x)$, both of **degree** d, which of the following statements correctly describes the necessary steps and constraints? [ ] C

Hint: Think about the number of points required to uniquely define the resulting product polynomial.

(A) We evaluate both polynomials at d points using FFT, perform pointwise multiplication in $O(d)$ time, and then use Inverse FFT to interpolate the coefficients.

(B) We evaluate both polynomials at $2d$ points using FFT, perform pointwise multiplication, and then use a standard $O(n^2)$ interpolation method to find the coefficients.

(C) we must evaluate both polynomials at n points where $n \geq 2d + 1$, perform pointwise multiplication of these values, and then use the Inverse FFT to interpolate the coefficients.

(D) The FFT allows us to multiply the coefficients directly in $O(d \log d)$ time without needing to convert the polynomials into point-value representation.

(b) **[4 marks]** In the FFT algorithm, why are the **complex roots of unity** used as the evaluation points rather than random real numbers? [ ] B

(A) They ensure that the resulting coefficients of the polynomial multiplication remain real numbers.

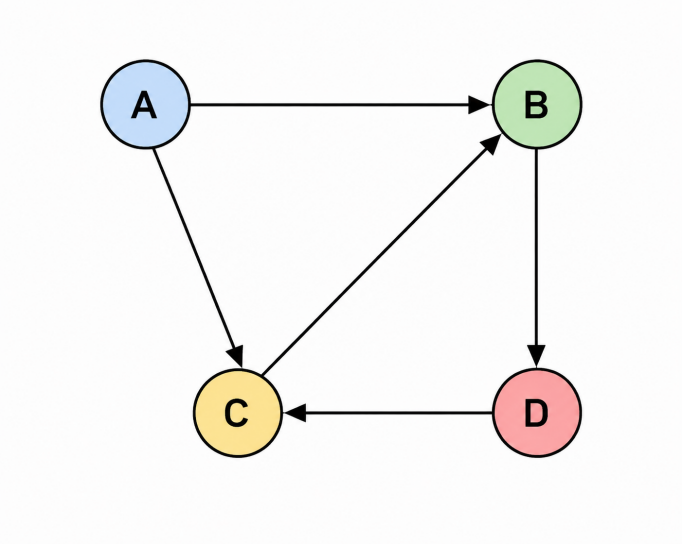
(B) Their mathematical properties (symmetry and periodicity) allow the problem to be recursively split into smaller sub-problems, reducing complexity from $O(n^2)$ to $O(n \log n)$.

(C) They are the only points that allow a polynomial of degree n to be uniquely represented by $n + 1$ values.

(D) They prevent the values of the polynomial from becoming too large (numerical overflow) during the evaluation step.

Question 6. Adjacency Matrix and Adjacency List **[15 marks]**

In the graph below, the nodes are labelled as A, B, C and D. The edges are all **directed**.



Please note that we use the labels (A-D), instead of integers(0-3), to represent the nodes in the following questions.

(a) **[4 marks]** Fill in the *adjacency matrix* of the above graph.

	A	B	C	D
A		1	1	
B				1
C		1		
D			1	

(Question 6 continued on next page)

(Question 6 continued)

(b) **[4 marks]** Construct an outgoing adjacency list representation, and an incoming adjacency list representation of the graph.

A: B, C
B: D
C: B
D: C

A:
B: A, C
C: A, D
D: B

(c) **[7 marks]** A social media company wants to store a graph representing friendships between users. The system contains 5 million users, and each user has approximately 200 friends on average.

Which data structure between *Adjacency Matrix* and *Adjacency List* has lower cost? Briefly justify your answer using memory efficiency, graph structure, and common graph operations..

Adjacency List.

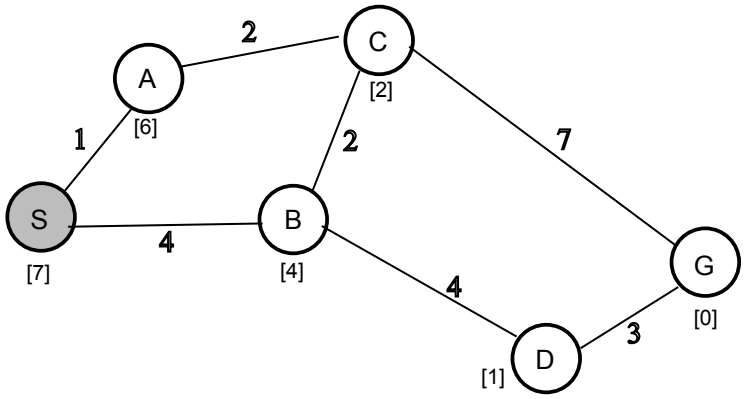
Matrix is 5m * 5m, adjacency list only store 5m *200

Very sparse graph

Need to find neighbours of a node quickly, for recommending friends etc.

Question 7. Shortest Paths **[30 marks]**

Consider the following graph. The nodes are the locations and the number on the edge (edge length) is the travel cost.



Heuristic values to goal node G are as follows (shown as the numbers in square brackets in the graph):

Node	h(n)
S	7
A	6
B	4
C	2
D	1
G	0

(a) **[10 marks]** A$*$ Search algorithm

Suppose we are using A$*$ Search algorithm to search for the shortest path from node S to node G in the graph below. The fringe is a priority queue of $\langle node, edge, CostSoFar, EstimatedTotalCost \rangle$ items, orderd by the EstimatedTotalCost.

Show what the algorithm will do on each of the next five iterations:

- which node it will visit
- what will be added to the Backpointers. If nothing is added, briefly explain why.
- what items will be added to the fringe. If nothing is added, briefly explain why. say 'done' when it is finished

Write the items and nodes in the format: $\langle node, edge, CostSoFar, EstimatedTotalCost \rangle$.

Iteration 1:

Node visited: $\langle S, null, 0, 7 \rangle$

Additions to Backpointers Map: None since S is the start node

Additions to fringe: $\langle$A, S-A, 1, 7$\rangle$ $\langle$B, S-B, 4, 8$\rangle$

Iteration 2:

Node visited: $\langle$A, S-A, 1, 7$\rangle$

Additions to Backpointers Map: add $\langle$A $\rightarrow$ S-A$\rangle$

Additions to fringe: $\langle$C, A-C, 3, 5$\rangle$

Iteration 3: (Say 'done' if finished)

Node visited: $\langle$C, A-C, 3, 5$\rangle$

Additions to Backpointers Map: add $\langle$C $\rightarrow$ A-C$\rangle$

Additions to fringe: $\langle$B, C-B, 5, 9$\rangle$ $\langle$G, C-G, 10, 10$\rangle$

Iteration 4: (Say 'done' if finished)

Node visited: $\langle$B, S-B, 4, 8$\rangle$

Additions to Backpointers Map: add $\langle$B $\rightarrow$ S-B$\rangle$

Additions to fringe: $\langle$D, B-D, 8, 9$\rangle$

Iteration 5: (Say 'done' if finished)

Node visited: $D, B-D, 8, 9\rangle$

Additions to Backpointers Map: add $\langle$D $\rightarrow$ B-D$\rangle$

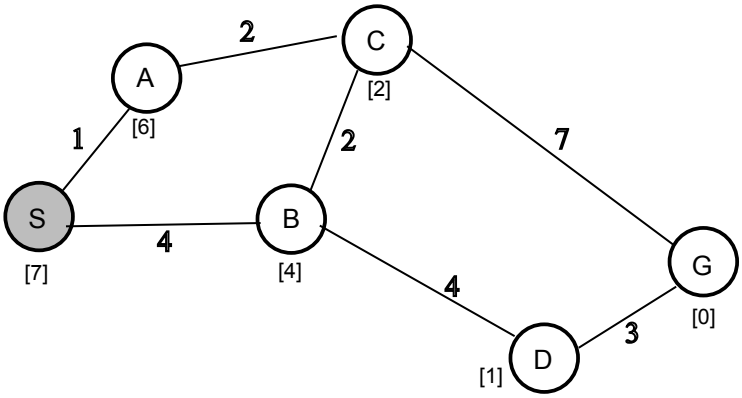
Additions to fringe: $\langle$G, D-G, 11, 11$\rangle$

(b) **[3 marks]** What is the path found using A$*$ Search algorithm?

S-A-C-G

(Question 7 continued on next page)

(Question 7 continued)



(c) **[8 marks]** What is path found if we are using a greedy best first search algorithm? Briefly explain the steps

S-B-D-G

Visit S, add S-A-6, S-B-4
Visit B, add B-C-2, B-D-1
Visit D, add D-G-0
Visit G

(Question 7 continued on next page)

(Question 7 continued)

(d) **[9 marks]** Real World Applications

You are designing a navigation system for:

1. Emergency ambulance routing (fastest possible guaranteed route is critical)
2. A video game enemy AI that needs to chase the player in real time
3. A GPS app that provides driving directions with traffic estimates

For each scenario:

- Choose the most appropriate algorithm among Dijkstra's, Greedy Best-First Search, and A*.
- Justify your choice in terms of speed, optimality, and use of heuristic information.

1 Emergency ambulance routing

Best choice: A*

Reason: needs optimal shortest-time route, heuristic (distance) improves efficiency while maintaining optimality.

2. A video game enemy AI that needs to chase the player in real time

Best choice: Greedy Best-First Search

Reason: speed is more important than optimal path; heuristic gives fast direction toward player.

3. A GPS app that provides driving directions with traffic estimates

Best choice: A*

Reason: balances optimal routing with heuristic guidance (distance/traffic estimates), practical efficiency.

* * * * * * * * * * * * * * *

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.