Question No. 1 [8 Marks]

Suppose that a Max-heap is implemented using an array, which is large enough to store all distinct words in a book. The heap is used to keep track of the frequency of the words in the book. The word that appears the most in the book has the highest frequency and is located at the root (array index 1). Each index of the array (heap) can store a word along with its frequency.
During input every word is inserted into the heap as follows: a word is read from the book and then the word is searched in the heap:

(i) if the word is not found it is then inserted into the heap with frequency 1.
(ii) if the word is found its frequency is increased by 1 and the heap is adjusted.

Answer the following questions.

a. **(2 Marks)** What is the worst-case running time, in big O notation, of searching for a word in the heap? Explain.
 1 point. O(n)
 1 point. Linear search is needed.

b. **(3 Marks)** Assume that a word has been searched and determined that it is not in the heap. What is the worst-case running time, in big O notation, of inserting a new word into the heap? Explain.
 1.5 points. O(1)
 1.5 points. A word is inserted with frequency 1 at the first empty array slot from left to right. The MAX-heap property is not violated because the parent word will not have a smaller frequency than 1.

c. **(3 Marks)** Assume that a word has been searched and determined that it is in the heap. What is the worst-case running time, in big O notation, to update a word's frequency? Explain.
 2 points. O(log n)
 1 points. Suppose that there are n words in the heap. All words have the same frequency. When a word at the leaf node has its frequency increased, it can move all the way to the root in the worst case.

Question No. 2 [7 Marks]

a) **(2 Marks)** Is an array that is sorted in ascending order a binary Min-heap? Why?
 Yes. An array in increasing order has the heap structural property as well as the heap order property (parent key <= children keys).

b) Below is an array representing a valid binary Min-heap.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Value	2	3	9	7	5	12	20	7	11	20	14	18		

(i) **(3 Marks)** Show the array contents after 2 insertions: first insert key **6** and then insert key **1**.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Value	1	3	2	7	5	9	6	7	11	20	14	18	12	20

(ii) **(2 Marks)** After the above insert operations, assume that two successive ExtractMin(HeapArray) operations are performed. Show the contents of the array after the operations.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Value	3	5	6	7	12	9	20	7	11	20	14	18		

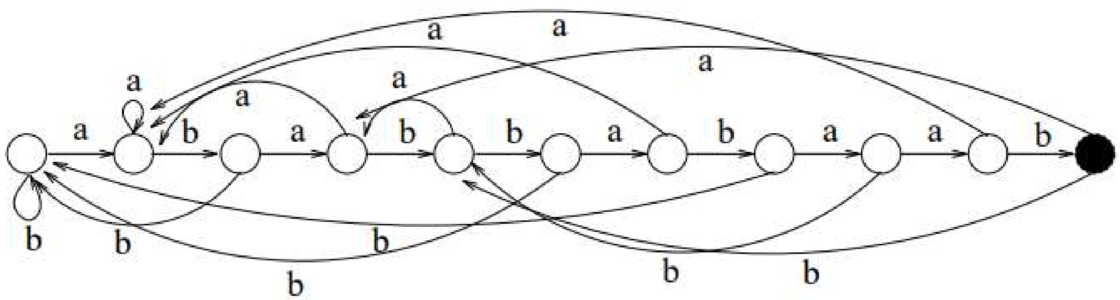
Question No. 3 [10 Marks]

In the following we only consider the binary alphabet $\sum = \{a , b\}$.

i. **(2 Marks)** Compute the Prefix function of pattern, P = ababbabaab using Knuth-Morris-Pratt algorithm for String matching.

I	1	2	3	4	5	6	7	8	9	10
P[i]	a	b	a	b	b	a	b	a	a	b
Π[i]	0	0	1	2	0	1	2	3	1	2

ii. **(4 marks)** Create Finite Automaton for the pattern P = ababbabaab.



iii. **(2 marks)** Compute the following considering P = abab

a. σ (acba) = **1**
b. σ (cbabab) = **4**

iv. **(2 marks)** We have seen the algorithms of Finite automaton and KMP for string matching that exploit the advantage of pre-processing on a string to reduce the string-matching time to *O(n)*. Which of the mentioned algorithms has benefit over the other? Justify your answer.

KMP algorithm avoids the computation of the transition function δ. It uses another function $\pi[1...m]$ pre-computed from the pattern P[1...m] in $\Theta(m)$ time. The prefix function π for a pattern holds knowledge about how the pattern matches against shifts of itself. This information can be used to avoid testing useless shifts that the naïve algorithm does. π contains only m entries, whereas δ is a table of $m|\sum|$ = md entries.

Question No. 4 [5 Marks]

Considering the brute-force (naïve) algorithm of String matching discussed in the class (and given below), how many character-comparisons are done to find all occurrences of the pattern ana in the text bananas and in the text anaeatsabanana?

For the text bananas, the brute-force algorithm does 9 comparisons in order to find all 2 occurrences of the pattern ana. For the text anaeatsabanana the brute-force algorithm does 21 comparisons in order to find all 3 occurrences of the pattern ana.

```
NAIVE-STRING-MATCHER(T, P)
n ← length[T]
m ← length[P]
for s ← 0 to n − m
    do if P[1..m] = T[s + 1..s + m]
          then print "Pattern occurs with shift" s
```

Question No. 5 [5 Marks]

The naive String matching algorithm, discussed in the class, only works for exact string matching. Your task is to rewrite the naive algorithm in such a way that if there is up to two mismatched characters found in the pattern at a shift s, the algorithm displays that the pattern is found. Otherwise, it must show that pattern is not found at the shift s.

```
Search (P,T)
BEGIN
        m ← Length (P), n ← Length (T), Count←0, j←0
        FOR (s ← 0 to n-m) DO
                FOR (j ← 1 to m) DO
                        IF (T[s + j] ≠ P[j]) THEN
                                Count ← Count + 1
                                IF (count > 2) THEN
                                        Break
                               END IF
                        END IF
                END FOR

                IF (j = m+1) THEN
                        PRINT "Pattern found at shift ", s
                ELSE
                        PRINT "Pattern not found at shift ", s
               END IF
        END FOR

END
```

Question No. 6 [15 Marks]

All parts carry equal marks. For the questions below, consider that V is the number of vertices and E is the number of edges in a graph.

A	Write the space complexity (using Big-O notation) of Adjacency List for storing a directed Graph.	O(V+E)
B	Write the time complexity (using Big-O notation) of BFS (Breadth First Search).	O(V+E)
C	If a graph has V vertices, what is the number of edges in its DF (depth first) spanning tree?	V-1
D	Why Prim's algorithm is categorized as a greedy algorithm?	Greedy choice: Extract node, having the least key, from the queue
E	Write the time complexity (using Big-O notation) of Dijkstra's algorithm if the queue (used in the algorithm) is a linear array.	$O(V^2+E)= O(V^2)$
F	Describe a greedy algorithm that does not always result in an optimal solution.	0-1 Knapsack problem; adding the item in sack that has the maximum worth (value to weight ratio)
G	What is Prefix constraint in Huffman Coding? Give an example.	One code cannot be a prefix of another, e.g., 0 and 01
H	Write the time complexity (using Big-O notation) of Huffman Coding algorithm if the queue (used in the algorithm) is a priority queue and C is the alphabet.	O(ClogC)
I	What is the minimum number of edges in a connected graph?	V-1
J	What is the maximum number of edges in a graph?	V(V-1)