

Assignment Two

Name	
Student number	

Direction:

Please answer all the questions below and hand in your answers before the due day. All work, must be handed in on time.

Due Date:

May 8, 2022.

Questions:

1. Calculate the time complexity of the following recurrence relation using the master theorem.

(a)

$$T(N) = 2T\left(\frac{N}{2}\right) + N\log N, \quad T(1) = 1$$

(b)

$$T(N) = 4T\left(\frac{N}{2}\right) + N, \quad T(1) = 0$$

(c)

$$T(N) = T\left(\frac{N}{2}\right) + 2^N, \quad T(1) = 1$$

2. Write the pseudocode of *merge sort*, and analyze the time complexity and space complexity.
3. Suppose there are six items with weights of (6, 2, 4, 3, 9, 12) and values of (9, 4, 6, 5, 14, 20). The capacity of the backpack is 16. You cannot break an item, either pick the complete item or don't pick it. Can you use *branch and bound method* to determine which items you should pick in order to maximize the value of the items without surpassing the capacity of your backpack? You should write the pseudocode (return the maximum value) and draw a solution space tree for this problem.

Assignment Two - Programming

Name	
Student number	

Direction:

Please answer all the questions below and hand in your answers before the due day. All work, must be handed in on time.

Due Date:

May 8, 2022.

Problem:

Please Write a program to solve the Sudoku puzzle. A sudoku solution should follow all of the following three rules:

1. Each of the digits 1-9 can only appear once in each row.
2. Each of the digits 1-9 can only appear once in each column.
3. Each of the digits 1-9 can only appear once in each of the 3x3 sub-boxes of the grid.

Tips:

- The space character indicates an empty cell.
- Puzzle board is a square matrix with a length of 9
- It is guaranteed that the input board has only one solution.

Core function:

```
void sudokuPuzzleSolver(vector<vector<char>>& board) {  
    // solve the sudoku puzzle by filling all the empty cells.  
}
```

Test Cases:

Case 1:

Input:

```
[["1", " ", " ", "4", "5", "6", "3", "7", "2"],
["7", "3", "6", "8", " ", " ", " ", " ", "5"],
[" ", "2", " ", "7", " ", "9", "1", " ", "8"],
["6", "4", " ", " ", " ", "8", "7", "5", " "],
[" ", "8", "2", " ", " ", "5", " ", " ", " "],
["5", " ", "1", "6", " ", "3", "8", " ", " "],
["3", "5", "7", "2", "9", "1", " ", "8", "4"],
[" ", " ", " ", "5", " ", "7", " ", " ", "3"],
["2", " ", " ", "3", "8", " ", " ", " ", " "]]
```

Output:

```
[["1", "9", "8", "4", "5", "6", "3", "7", "2"],
["7", "3", "6", "8", "1", "2", "9", "4", "5"],
["4", "2", "5", "7", "3", "9", "1", "6", "8"],
["6", "4", "3", "9", "2", "8", "7", "5", "1"],
["9", "8", "2", "1", "7", "5", "4", "3", "6"],
["5", "7", "1", "6", "4", "3", "8", "2", "9"],
["3", "5", "7", "2", "9", "1", "6", "8", "4"],
["8", "1", "4", "5", "6", "7", "2", "9", "3"],
["2", "6", "9", "3", "8", "4", "5", "1", "7"]]
```

Explanation:

1			4	5	6	3	7	2	1	9	8	4	5	6	3	7	2
7	3	6	8					5	7	3	6	8	1	2	9	4	5
	2		7		9	1		8	4	2	5	7	3	9	1	6	8
6	4				8	7	5		6	4	3	9	2	8	7	5	1
	8	2			5				9	8	2	1	7	5	4	3	6
5		1	6		3	8			5	7	1	6	4	3	8	2	9
3	5	7	2	9	1		8	4	3	5	7	2	9	1	6	8	4
			5		7			3	8	1	4	5	6	7	2	9	3
2			3	8					2	6	9	3	8	4	5	1	7

Keypoints:

- You can choose any programming language to implement this problem.
- You should use **backtracking** to solve the problem
- Your program should run successfully, output the correct answer, and provide **screenshots** in the document.
- Please make sure there are necessary comments in your source code. **Plagiarism is strictly forbidden!**

Submission:

- Source codes without project files.
- A brief documentation (PDF is recommended), including algorithm idea, algorithm complexity analysis, and screenshots of running results.
- Pack all above files **with the answer of the calculation questions**, and compress them into a **ZIP** file. Please rename the **ZIP** file as '**StudentID_Name_Assignment_2.zip**'.
- Send the zip file to the email:
 - **1012376712@qq.com**
- Please send the email by **May.8, 2022**.