TASK SHEET 5

1. Write a program that efficiently finds the maximum sum subarray within a given list of integers.
2. Implement an algorithm to efficiently find the longest increasing subsequence in a list of integers.
3. Develop a function that efficiently computes the square root of a given number without using built-in functions.
4. Create a program that efficiently checks if a given string contains all unique characters.
5. Design a function that efficiently sorts a list of strings based on their lengths.
6. Write a program that efficiently determines the minimum number of coins needed to make a given amount of change.
7. Implement an algorithm to efficiently solve the traveling salesman problem using dynamic programming.
8. Develop a function that efficiently computes the power set of a given set.
9. Create a program that efficiently checks if a given sudoku puzzle is solvable.
10. Design a function that efficiently determines if a given string is an anagram of another string.
11. Write a program that efficiently computes the intersection point of two linked lists.
12. Implement an algorithm to efficiently find the kth smallest element in a binary search tree.
13. Develop a function that efficiently converts a Roman numeral to an integer.
14. Create a program that efficiently checks if a given graph is bipartite.
15. Design a function that efficiently finds the longest palindrome substring within a given string.