Problem Set: ECS 202 Data Structures and Algorithms

Last Updated: April 14, 2024

---

1. Define data structure and given examples of primitive data structure.
2. Write a program that takes as input an integer n, followed by an array of n integers, and outputs a maximum element in it.
3. Write a program that takes as input an integer n, followed by an array of n integers, and outputs a maximum possible product of two elements in it.
4. Write a program that takes as input an integer n and then outputs n^{th} Fibonacci number.
5. For a function $g(n) : \mathbb{N} \mapsto \mathbb{N}$, define sets $\mathcal{O}(g)$, $\Omega(g)$, $\Theta(g)$, $o(g)$, and $\omega(g)$. Give examples of at least two functions in each of this set.
6. In each of the following situations, indicate whether $f \in \mathcal{O}(g)$, or $f \in \Omega(g)$, or $f \in \Theta(g)$, or $f \in o(g)$, or $g \in w(g)$.

   (a) $n - 100$, $n - 200$
   (b) $n^{1/2}, n^{2/3}$
   (c) $100n + \log n, n + (\log n)^2$
   (d) $n \log n$, $10n \log(10n)$
   (e) $\log(2n), \log(3n)$
   (f) $10 \log n$, $\log(n^2)$
   (g) $n^{1.01}$, $n \log^2 n$
   (h) $n^2/\log n$, $n(\log n)^2$
   (i) $n^{0.1}$, $(\log n)^{10}$
   (j) $(\log n)^{\log n}$, $n/\log n$
   (k) $\sqrt{n}, (\log n)^3$
   (l) $n^{1/2}, 5^{\log_2 n}$
   (m) $2^n, 3^n$
   (n) $2^n, 2^{n+1}$
   (o) $n!, 2^n$
   (p) $(\log n)^{\log n}, 2^{(\log_2 n)^2}$

7. Write a program that takes as input an integer n, followed by an array of characters R, B, and Y. You program should compute minimum number of swaps needed to arrange the array such that all Bs appear before all Ys which appear before all Rs.

   (Reference Chapter 4.3 in [1])
8. Write a pseudo code for (i) bubble sort, (ii) linear search, and (iii) binary search. In each case, specify input and output.
9. Write a program that takes as input an integer n, followed by an array of n integers and sorts it using bubble-sort. Your code should output the number of swaps for given input.
10. Write a program that takes as input an integer D, followed by two arrays of size $D + 1$. Each array represents a polynomial of degree at most D, where i^{th} entry in the array is the co-efficient of x^i in the polynomial. Your program should compute the product of these two polynomial. The final output should be the sum of coefficients in the resulting polynomial.
11. Write a program to read an array of n numbers and then find the smallest number *using pointers*.

12. Write a program to interchange the largest and the smallest number in an array *using pointers.*

13. Write a program to read, display, add, and subtract two complex numbers *using structures.*

14. Declare a structure that represents the following: information. Roll Number, Name, Gender, Date of Birth Marks in English, Marks in Mathematics, and Marks in Computer Science.

15. Write a program to add and subtract height 6'2" and 5'4".

16. Write a program to find whether a number is even or odd using functions.

17. Write a program to add two integers and display their output using pointers and functions.

18. Write a program to print all prime numbers from m to n using function.

19. Write a program to read numbers until –1 is entered and display all the numbers.

20. Write a program that performs the following tasks.

    (a) Create linked list
    (b) Display all the items in the linked list
    (c) Insert a node at the beginning
    (d) Insert a node at the end
    (e) Insert a node before a node with specific value
    (f) Insert a node after a node with specific value
    (g) Delete the node at the beginning
    (h) Delete the node at the end
    (i) Delete the node with specific value.
    (j) Delete the node that comes after a specific value
    (k) Delete the entire list (and free memory space)
    (l) Sort the linked list (using bubble sort).

21. Write a program to implement a stack using linked list.

22. Write a program to check nesting of parentheses using a stack.

23. Write a program to convert an infix expression into its equivalent postfix notation.

24. Write a program to implement a circular queue.

25. Write a program to implement a priority queue.

26. Write a program that constructs a binary search tree and supports the following operations. You can assume that you are working only with positive integers and the input values are unique.

    (a) Insert Element
    (b) Preorder Traversal
    (c) Inorder Traversal
    (d) Postorder Traversal
    (e) Find the smallest element
    (f) Find the largest element
    (g) Delete an element
    (h) Count the total number of nodes
    (i) Count the total number of leaves
    (j) Count the total number of internal nodes (i.e. nodes that are not leaves)
    (k) Determine the height of the tree
    (l) Find the mirror image of the tree
    (m) Delete the tree

27. Write a program that constructs an AVL tree and supports the operations mentioned above.

28. Write a program that constructs a Red-Blue tree and supports the operations mentioned above.

29. Write a program that constructs a binary heap and supports the operations of inserting and deleting.

30. Write a program that reads a list of edges and and constructs both adjacency list representation and adjacency matrix representation of the graph. You can assume that the input graph contains at most 200 vertices.

31. Write a program that computes the following quantities for a graph given in adjacency list representation.

    (a) Number of edges.
    (b) Highest Degree.
    (c) Lowest Degree.
    (d) The number of vertices with highest degree.
    (e) Number of connected components.

32. Write a program to implement the breadth-first search algorithm.

33. Write a program to implement the depth-first search algorithm.

34. Write a program which finds the cost of a minimum spanning tree.

35. Write a program to enter n numbers in an array. Redisplay the array with elements being sorted using the bubble sort in descending order.

36. Write a program to enter n numbers in an array. Redisplay the array with elements being sorted using the insertion sort in descending order.

37. Write a program sorts an array of n integers using recursively implementation of merge sort.

38. Write a program sorts an array of n integers using recursively implementation of quick sort.

39. Write a program to implement shell sort algorithm.

40. Write a problem to solve Scheduling Events problem discussed in Lecture 33.

41. Write a problem to solve Scheduling Events problem discussed in Lecture 33.

42. Write a problem to solve Collection of Coins problem discussed in Lecture 34.

References

[1] Seymour, Lipschutz, *Data Structures with C*, Publisher: Tata McGraw-Hill