

**Savitribai Phule Pune University**

**Department of Scientific Computing , Modeling and Simulation**

**End Semester Examination Feb. 2023**

**Course No. : SC – 101**

**Title : Programming Languages and Principles - I**

**Date : 06/02/2022**

**Time : 2.5 hrs (2:30pm to 5:00pm) Marks : 50**

**Q1 Answer the Following.1 mark each. (10 marks)**

**Instruction:** Write only the alphabets and not the actual text given. Write ALL applicable

Options.

1. What is the right sequence for the following activities? (Write the alphabets in the

right order – such as ABCDE)

A. Understanding the problem B. Writing code for the problem

C. Writing the specification D. Writing the algorithm

E. Testing the Application

1. From the following, which all are not the criteria for an algorithm? (write the

alphabets for all the ones that are not criteria)

A. Definiteness B. Effectiveness C. Vastness D. Finiteness E. Goodness

1. For the following function declaration, which all are part of the invariable part of the

space requirement?

Void MyFunNew(int iCount, char cStr, long lArr[ ], float fMultFact)

A. iCount B. cStr C. lArr D. MultFact

4. Write the following asymptotic complexities in the increasing order assuming n > 100?

(write the alphabets in the right order such as ABCD)

A. nlogn B. n C. nlogn D. n2

5. Which of the following is not true for Dynamic programming?

1. Applied to constraint satisfaction problems
2. Values are computed Recursively
3. Optimum solution is computed in bottoms up fashion
4. Results of sub problems are stored to be used later

6. Which of the following is not a divide & conquer algorithm?

A Quick Sort B Bellman-Ford Algorithm C Binary Search D Merge Sort

7. Which of the following algorithm’s performance will improve by use of the HEAP data

structure?

1. Selection Sort
2. Bellman-Ford Algorithm
3. Dijkstra’s Algorithm
4. Fractional knapsack

8. What is the worst case time complexity of this code segment?

int fun(int n, int A[])

{

int i, j, count = 0;

for (i = 0; i < n; i++)

{

j = 0;

while (j < n && A[i] <= A[j])

j++;

count += j;

}

return count;

}

9. For a simple, undirected, connected graph with 6 vertices and 5 leaves, what is the

highest degree a vertex will have?

10. What is the recurrence Relation for Merge sort?

**Q2. State if each of the following problem/algorithm is in P or NP (03)**

1. Matrix Multiplication
2. Insert a number in sorted list
3. Node/Vertex Cover
4. Greedy Algorithm
5. Subset Sum
6. Dijkstra’s shortest path algorithm

**Q3. For the following problems, between branch & bound and backtracking, (02)**

**which would be appropriate?**

1. Traveling Salesman
2. Subset Sum
3. 0-1 Knapsack
4. 8 Queens problem

**Q4. Solve any three of the following.(5 marks each.) (15)**

1. Write the Insertion sort algorithm and work out it’s time complexity using the table method
2. Write the merge sort algorithm and explain what is it’s recurrence equation and what the time complexity works out to be
3. Write the Bellman-Ford algorithm for the shortest path and explain how it identifies a graph with negative weight cycle
4. Write Non deterministic algorithm for Search (2 marks)

Write Non deterministic algorithm for Sort (3 marks)

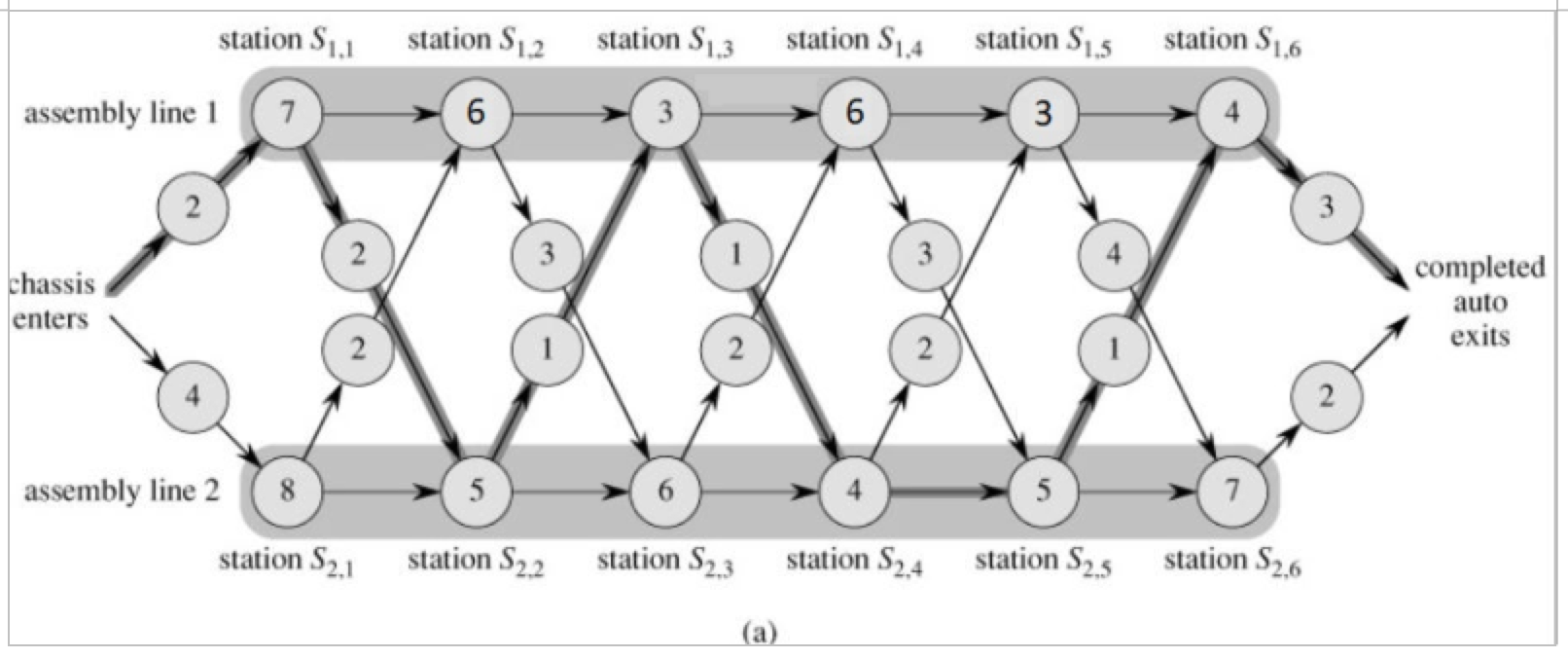
**Q5. Solve any one of the following. (10)**

1. A Knapsack with the maximum capacity given in kgs needs to be filled with items selected from the given list of items. For each item its weigh in kgs and its cost price is given. It can be assumed that all of them can be sold for the same price X. Write the algorithm to maximize the profit. (3)
   1. Apply this algorithm to the following set of items. The capacity of the Knapsack is 30 kgs. And the price at which each item will be sold is Rs. 50. Explain what items will be selected and in what order. Will there be any partial item selected? If yes, what would be the fraction of which item? (2)
   2. What is the asymptotic complexity of this algorithm? (2)
   3. Explain what would need to change in the algorithm if the values given under cost were actually profits and not the costs associated with the items. (1)
   4. Based on the condition given in (iii) above, what items will be selected? (2)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Items No** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Weight** | 8 | 8 | 12 | 20 | 5 | 2 | 15 | 6 | 14 | 10 |
| **Cost** | 13 | 17 | 12 | 45 | 3 | 5 | 5 | 6 | 6 | 10 |

OR

1. Write the fastest line (fastest way out) algorithm for the assembly line problem.
2. The following figure shows the assembly line with the fastest way out marked with a thick black line. However, some of the stations on line 1 have been upgraded recently and the times they take have changed. So using your algorithm, write the values of f\* (fastest way through stations on the two lines) for all the stations.
3. The fastest way highlighted by the thick line is **“enter L1 S(1,1) S(2, 2) S(1, 3) S(2, 4) S(2,5) S(1,6) exit L1”**. With the updated stations of line 1, what would be the fastest way now? You can write in the same fashion.



**Q6. Solve any one of the following. (10)**

Even if you can’t reach up to the algorithm, write down what you have understood and your thoughts on HOW the problem can be solved.

1. Given Text in 2D Matrix T[m][n], and a pattern P[k], you need to search for P in T. One is allowed to move only down or right while searching for the pattern.

Example:

[[a, b, c],

[d, e, c],

[e, f, r]]

Pattern “abefr” is present. But the pattern “abefe” is not present.

Write the algorithms for finding the pattern. Does this fall into any known category of algorithms? What is the time complexity?

Explain any assumptions you are making and the logic of the solution.

OR

1. An edge in an undirected connected graph is a bridge iff, removing it disconnects the graph.
2. Given an undirected connected Graph G(V,E) how can you find out the bridges in the graph? Explain the method you will use.
3. Given an edge (u,v), write the algorithm to check if it is a bridge.
4. Work out the complexity of your algorithm

OR

1. Given an expression string exp, write the algorithm to examine whether the pairs and the orders of the different parentheses are matched perfectly (that is {, } and ( , ) and [ , ]) in exp. What is the time & space complexity of this algorithm?

\*\*\*\*\*