Computer Algorithm

Tutorial No.- 3 Unit II

1. Explain
    a. Feasible Solution
    b. Objective Function
    c. Optimal Solution
2. Compare Subset Paradigm with ordering paradigm
3. Give solution to Knapsack Problem using greedy method
4. Example 1 on Knapsack Problem

    n=3 m=20

    $(p_1, p2, p3) = (25,24,15)$

    $(w_1, w2, w3) = (18, 15, 10)$

5. Example 2 on Knapsack Problem

    n=7 m=15

    (p1, p2, p3, p4, p5, p6, p7) = (10,5,15,7,6,18,3)

    $(w_1, w_2, w_3, w_4, w_5, w_6, w_7) = (2,3,5,7,1,4,1)$

6. Example 3 on Knapsack Problem

    n=4 m=25

    $(p_1, p2, p3, p_4) = (2, 5, 8, 1)$

    $(w_1, w2, w3, w_4) = (10, 15, 6, 9)$

7. Example 4 on Knapsack Problem

    n=6 c=20

    (p1, p2, p3, p4, p5, p6) = (12, 5, 15, 7, 6, 18)

    $(w_1, w_2, w_3, w_4, w_5, w_6) = (2, 3, 5, 7, 1, 5)$

8. Example 5 on Knapsack Problem

    n=4 m=30

    $(p_1, p2, p3, p_4) = (27, 20, 24, 15)$

    $(w_1, w2, w3 , w_4) = (15, 10, 18, 10)$

9. Prove that if P1/W1 >=P2/W2 >= …> Pn/Wn. Then greedy method generates an optimal solution to the given instance of knapsack problem

10. Give solution to "job sequencing with deadlines" using greedy method.

11. Example 1 on Job sequencing with deadline

    n=4

    $(p_1, p_2, p_3, p_4) = (100, 10, 15, 27)$

    $(d_1, d_2, d_3, d_4) = (2, 1, 2, 1)$

12. Example 2 on Job sequencing with deadline

    n=5

    $(p_1, p_2, p_3, p_4, p_5) = (20, 15, 10, 5, 1)$

    $(d_1, d_2, d_3, d_4, d_5) = (2, 2, 1, 3, 3)$

13. Example 3 on Job sequencing with deadline

    n=7

    (p1, p2, p3, p4, p5, p6, p7) = (3, 5, 20, 18, 1, 6, 30)

    $(d_1, d_2, d_3, d_4, d_5, d_6, d_7) = (1, 3, 4, 3, 2, 1, 2)$

14. Example 4 on Job sequencing with deadline

    n=5

    $(p_1, p_2, p_3, p_4, p_5) = (45, 15, 20, 7, 65)$

    $(d_1, d_2, d_3, d_4, d_5) = (1, 3, 2, 1, 2)$

15. Example 5 on Job sequencing with deadline

    n=7

    (p1, p2, p3, p4, p5, p6, p7) = (50, 15, 18, 16, 8, 25, 60)

    $(d_1, d_2, d_3, d_4, d_5, d_6, d_7) = (1, 3, 4, 3, 2, 1, 2)$

16. Example 6 on Job sequencing with deadline

    n=5

    $(p_1, p_2, p_3, p_4, p_5) = (20,16,11,5,25)$

    $(d_1, d_2, d_3, d_4, d_5) = (2,2,1,2,1)$

17. Example 7 on Job sequencing with deadline

    n=7

    (p1, p2, p3, p4, p5, p6, p7) = (45,5,20,18,6,30,70)

    (d_1, d_2, d_3, d_4, d_5, d_6, d_7) = (1,3,4,3,2,1,2)

18. Explain the following

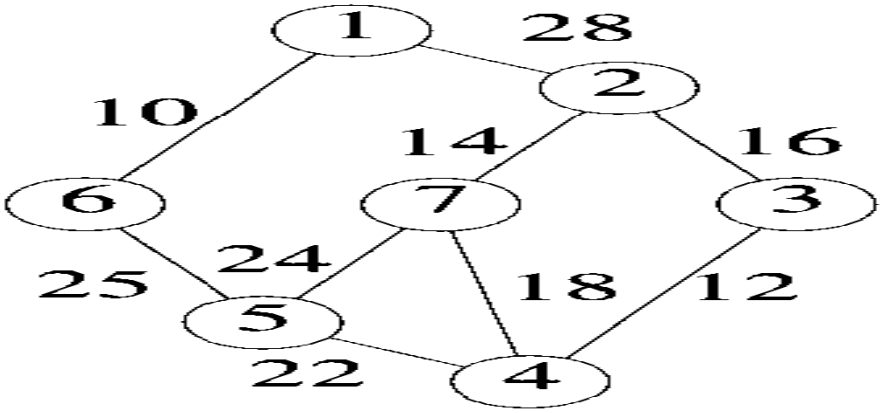
    a. Spanning Tree
    b. Minimal Spanning Tree
    b. Weighted Graph

19. Explain Prim's algorithm to find Minimum Cost Spanning Tree (MST)

    Explain Kruskal's algorithm to find Minimum Cost Spanning Tree (MST)

    Compare Prim's and Kruskal's algorithm to find Minimum Cost Spanning Tree (MST)

20. Find Minimum Cost Spanning Tree using Prim's and Kruskal's algorithm in following graph



(a)

21. Give applications of MST
22. Give solution to optimal storage on tapes problem using Greedy Method
23. Give different solutions to Optimal Merge Pattern problem using Greedy Method. State which is better
24. Give applications of MST
25. Give solution to optimal storage on tapes problem using Greedy Method

    Write note on optimal storage on tapes problem

26. Discuss the solution of extension of optimal storage on tape problem to several tapes

27. Give different solutions to Optimal Merge Pattern problem using Greedy Method. State which is better
28. Give applications of Optimal Merge Pattern
29. Find an optimal merge pattern for ten files whose lengths are 26, 32, 12, 6, 85, 55, 90, 35, 3 and 11
30. Find an optimal merge pattern for ten files whose lengths are 28, 32, 12, 5, 84, 53, 91, 35, 3 and 11
31. Write note on "Huffman Code"
32. Obtain a set of optimal Huffman codes for the messages (M1,…,M7) with relative frequencies (q1,…,q7)=(4, 5, 7, 8, 10, 12, 20). Draw the decode tree for this set of codes.
33. Obtain a set of optimal Huffman codes for seven message with relative frequencies (3, 5, 9, 13, 21, 25, 30)
34. Give solution to single source shortest path problem using Greedy Approach
    Write note on single source shortest path problem
35. Find shortest path of all the vertices from single source vertex A using Greedy Approach.