Assignment 5

COMP 4500

Due: March 22, 2024 by 11:59 p.m.

Upload your Word, PDF, or LaTeX file(s) to Blackboard

Exercises from the Textbook (50 points):

Page 312:

1(a,b)	(10 points)
2	(20 points)
6	(10 points)
7	(10 points)

Additional Exercises (50 points):

1. Consider a set ***F*** of friends. An **essential loser** is a person ***p*** in ***F*** who knows something about everyone else in ***F*** yet exactly one other person knows something about ***p***.
    a. How many essential losers can there be in a set of friends? Explain.
    b. Given a set ***F*** of size larger than the answer given in (a), how can you find at least one person who is *not* an essential loser in constant time? Note that you do not have a graph representation of set ***F***. Instead, you have a black-box function ***knows***(p_1, p_2) that will return true if and only if p_1 knows something about p_2. This function operates in constant time. Also, you can remove a random element from set ***F*** in constant time.
    c. Explain how to use the answer to part (b) to design an efficient algorithm to find all the essential losers in a set of ***n*** friends. Be sure to give the running time.

2. Draw the two-dimensional table that shows the dynamic programming solution for optimal sequence alignment between the strings “breakfast” and “craziest”. How much does the optimal sequence alignment cost? The cost of an insertion (or deletion) δ is 1. The cost of replacing any letter with a different letter is 1. The cost of replacing any letter with itself is 0.

3. Draw the two-dimensional table corresponding to the knapsack problem for the following objects:
    a. (80, 9)
    b. (93, 2)
    c. (99, 2)
    d. (36, 2)
    e. (5, 9)
    f. (37, 10)
    g. (2, 9)
    h. (81, 14)
    i. (44, 11)
    j. (68, 6)

    Each object is given in the form (v_i, w_i) where v_i is the value of object *i* and w_i is the weight of object *i*. The knapsack can hold a maximum weight of 20. Which objects are included in the maximum value collection?

4. Consider the following weighted intervals:
    a. (14, 19, 18)
    b. (25, 32, 8)
    c. (34, 35, 17)
    d. (7, 37, 8)
    e. (19, 67, 19)
    f. (68, 69, 20)
    g. (55, 78, 19)
    h. (53, 79, 14)
    i. (58, 81, 6)
    j. (55, 85, 15)
    k. (38, 88, 9)
    l. (90, 92, 12)
    m. (64, 111, 8)
    n. (69, 115, 1)
    o. (92, 139, 18)

    Each interval is given in the form (s_i, e_i, w_i) where s_i is the start time of interval *i*, e_i is the end time of interval *i*, and w_i is the weight of interval *i*. Note that the intervals have already been sorted by ending time. Give the array of subproblem values showing the optimal solutions from the one using no intervals up to the one that might use all 15. Then, state which intervals belong in the optimal solution. It is permitted for two intervals to be in the solution if one ends at the same instant that the other starts.

5. You are working for the CIA, smuggling plutonium out of the former Soviet Union so that it doesn't fall into the hands of terrorists. You have a special plutonium carrying case with weight capacity ***W***. However, the carrying case also has a radiation capacity ***R***. You stumble into an abandoned nuclear launch site and find many chunks of plutonium. Each chunk has a weight ***w*** and radiates a steady amount of radiation ***r***. Can you find a set of plutonium chunks whose weight sums to exactly ***W*** and whose radiation sums to exactly ***R***?
    a. State what the subproblems are for this problem. Which of these subproblems is the full problem? You should use a record-keeping system similar to the OPT(***i***,***w***) values used for subset sum.
    b. Explain which subproblems are so easy to solve that the solution is known immediately. For the harder subproblems, describe their solutions in terms of other subproblems.
    c. Formulate these relationships into a full algorithm for the problem. Be sure to give its running time and space requirements. **Hint:** A two-dimensional table might not be sufficient.