19CSE302: Design and Analysis of Algorithms
Tutorial 2: Part 1
15.11.2023

1. Suppose you are organizing a party for a large group of your friends. Your friends are pretty opinionated, though, and you don't want to invite two friends if they don't like each other. So you have asked each of your friends to give you an "enemies" list, which identifies all the other people among your friends that they dislike and for whom they know the feeling is mutual. Your goal is to invite the largest set of friends possible such that no pair of invited friends dislike each other. How would you solve this problem. Model this problem appropriately and explain the model.
    1. Give a greedy algorithm for the solving the problem. Prove that it is optimal or give a counter example.
    2. Can this be solved using DP or Backtracking. Explain your answer and give a suitable algorithm.
    3. Implement the two techniques and give test cases to demonstrate the different cases

2. Consider the problem of finding a maxima set for a set, S, of n points in the plane. A point is a maximum point in S if there is no other point, (x', y'), in S such that $x \leq x'$ and $y \leq y'$. Points that are not members of the maxima set can be eliminated from consideration, since they are dominated by another point in S. Thus, finding the maxima set of points can act as a kind of filter that selects out only those points that should be candidates for optimal choices.
    1. What design strategy will you use to solve this problem and why?
    2. Give the algorithm and demonstrate its correctness and calculate the running time complexity.
    3. Implement this algorithm and analyse the working of the algorithm by tracing intermediate steps. Also show the effectiveness of the algorithm through test cases.

3. Implement the solution to the n-queen's problem using backtracking approach. The implementation must clearly model the state space tree. Trace the state space tree and run the algorithm till you find 3 potential solution configurations. Test your algorithm for minimum n=4 and maximum n=8.

4. Suppose that three workers are given the task of scanning through a shelf of books in search of a given piece of information. To get the job done fairly and efficiently, the books are to be partitioned among the three workers. To avoid the need to rearrange the books or separate them into piles, it is simplest to divide the shelf into three regions and assign each region to one worker. But what is the fairest way to divide up the shelf? If all books are the same length, the job is pretty easy. Just partition the books into equal-sized regions,

    100 100 100 | 100 100 100 | 100 100 100

    so that everyone has 300 pages to deal with. But what if the books are not the same length Suppose we used the same partition when the book sizes looked like this:

    100 200 300 | 400 500 600 | 700 800 900

    I, would volunteer to take the first section, with only 600 pages to scan, instead of the last one, with 2,400 pages. The fairest possible partition for this shelf would be

    100 200 300 400 500 | 600 700 | 800 900

    where the largest job is only 1,700 pages and the smallest job 1,300. In general, we have the following problem:

    Problem: Integer Partition without Rearrangement
    Input: An arrangement S of nonnegative numbers {s1, . . . , sn} and an integer k.
    Output: Partition S into k or fewer ranges, to minimize the maximum sum over all the ranges, without reordering any of the numbers.
    1. Give an efficient algorithm to solve this problem. What design technique would you use and why. Explain the details of the algorithm with correctness and time complexity.
    2. Implement the solution and demonstrate the working of your algorithm and show the correctness using appropriate test cases.