CS2020: Data Structures and Algorithms (Accelerated)

Problems 1–2

For Week13

1 Lecture 21 Material - The True Form of DP

1.1 1-D Max Range Sum Problem - solution that is faster than $O(N^2)$

There is a 'greedy' solution for this problem that runs in $O(N)$. Do quick searches in the Internet using related terms 'max sum', 'maximum contiguous sum', 'maximum subarray problem', etc... One student will be selected to present his/her findings in DG.

1.2 2-D Max Range Sum Problem - $O(N^4)$ or $O(N^3)$ solution

The Max Range Sum/max subarray problem can also be asked on a 2-D array (matrix). For simplicity, let's assume the we have a square matrix of size $N \times N$. A naive solution is to try all possible top-left and bottom-right coordinate of the sub array (sub matrix) and then do another two nested loops again to compute the sum of that sub matrix. This is $O(N^6)$. Now your task is to solve this problem using either $O(N^4)$ or $O(N^3)$ algorithm.

1.3 Subset Sum Problem - the other form of 0-1 knapsack problem

The subset sum problem is this: you are given a set of integers S and a target integer V, is there a non empty subset of S that sum to V?

This problem is very similar to the 0-1 knapsack problem and thus has similar DP solution. Use your knowledge of the DP solution of 0-1 knapsack problem to solve the following problem.

Suppose you are given this list of integers = {2, 3, 4, 1, 2, 5, 10, 50, 3, 50}. You have to decide if you can split these numbers into two disjoint subsets such that the sums of both subsets are the same, i.e. 130 / 2 = 65. The answer is YES, it is possible, one of the solution is: Subset1 = {5, 10, 50}, Subset2 = {2, 3, 4, 1, 2, 3, 50}.

Write a DP solution for the general version of this problem. Given a list of n positive integers where the total sum of all n integers is V, is it possible to split the n integers into two sets that both sum to $V/2$? It is known that $1 \leq n \leq 100$, $2 \leq V \leq 1000$.

1.4 String Alignment → Longest Common SUBSEQUENCE

In Lec11, we have seen the Longest Common SUBSTRING problem between two strings. The solution presented in that lecture uses hashing (rolling hash) that runs in $O(n \log n)$. (Do you still remember this?).

Now we are interested in the Longest Common SUBSEQUENCE problem, a problem to find the longest subsequence common to both sequences (in this case, string).

Note that 'subsequence' (recall the Longest Increasing Subsequence from Lec18) is different from a 'substring', for example, given a string S = "STEVEN", then A = "SEVEN" is a subsequence (but not a substring) of S, and B = "EVEN" is a substring (and also a subsequence) of S.

This Longest Common Subsequence problem can be solved by modifying the $O(nm)$ Needleman-Wunsch Dynamic Programming algorithm presented in Lec21 as the solution for the String Alignment problem. Find a similarity between the String Alignment problem and the Longest Common Subsequence problem, then use it to modify small part of the Needleman-Wunsch DP algorithm.

2 Identifying Problems Solvable with DP

For each of the following scenario, answer these: is this problem has 'optimal sub-structures' and 'overlapping sub-problems' which can be solved with DP technique? If yes, what is the DP formulation (distinct states, space complexity, overlapping transitions, time complexity)? If no, what should be used to solve it (this also serves as a semester review)?

- Sorting a set of integers S. Member of S are integers from this range [1..100].
- Searching for an integer in an unsorted integer array.
- Deciding if a given graph is 2-colorable (recall one of the quiz 2 problem).
- Counting how many times a word of length Q (say, few characters) appears in a long string (imagine DNA data) of length N (say, 1 million characters).
- Deciding if we can pay X SGD ($X < 10$ SGD) using the limited coins that we have (a 1 dollar coins, b 50 cents coins, c 20 cents coins, d 10 cents coins, and e 5 cents coins).

3 Semester Review

If you still have time, you can spend some time during the last DG10 to review the long semester.

And as PS10 deadline has been extended to Friday 15 April 2011, 9am. You can use this opportunity to discuss the problems if you still have difficulties.

Finally, you may also want to spend some time to take a DG photo as a momento and upload the photo to your Facebook :).