**Advanced Algorithms**

**Exercise for Lecture 11**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** |  | **Student ID** |  |
| **Problem 1** |  | | |
| **Problem 2** |  | | |
| **Problem 3** |  | | |
| **Total Score** |  | | |
| **Notes** | Deadline: **2023-10-20 24:00**  Submission Format: ‘**Lecture11\_Name\_Student ID.docx**’, and please send to: **[algorithms\_23fall@163.com](mailto:algorithms_23fall@163.com)**.  This assignment is meant to be an evaluation of your **individual** understanding coming into the course and should be completed **without collaboration** or outside help. | | |

**Problem 1. [20 points]** Determine an LCS of and using the following table.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0 |  |  |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |  |  |
|  | 0 |  |  |  |  |  |  |  |  |  |

**Solution:**

**Problem 2. [35 points]** Write an algorithm MAXIMAL-NON-ADJACENT-SEQUENCE-WEIGHT() that, given a sequence of numbers , computes, with worst-case complexity , the maximal weight of any sub-sequence of non-adjacent elements in . A sub-sequence of non-adjacent elements may include or but not both, for all . For example, with , MAXIMAL-NON-ADJACENT-SEQUENCE-WEIGHT() should return .

**Solution:**

**Problem 3. [45 points]** Santa Claus is packing his sleigh with gifts. His sleigh can hold no more than pounds. He has different gifts, and he wants to choose a subset of them to pack in his sleigh. Gift has utility (the amount of happiness gift induces in some child) and weight . We define the weight and utility of a set of gifts as follows:

* The weight of a set of gifts is the **sum** of their weights.
* The utility of a set of gifts is the **product** of their utilities.

For example, if Santa chooses two gifts such that , and , , then the total weight of this set of gifts is pounds and the total utility of this set of gifts is . All numbers mentioned are positive integers and for each gift , . Your job is to devise an algorithm that lets Santa maximize the utility of the set of gifts he packs in his sleigh without exceeding its capacity .

1. Give a recurrence that can be used in a dynamic program to compute the maximum utility of a set of gifts that Santa can pack in his sleigh. Remember to evaluate the base cases for your recurrence.
2. Write pseudo-code for a dynamic program that computes the maximum utility of a set of gifts that Santa can pack in his sleigh. What is the running time of your program?
3. Modify your pseudo-code in part *b* to output the actual set of gifts with maximum utility that Santa packs in his sleigh.

**Solution:**