## Lab #5: Dynamic Programming

Students implemented the following problems using "Dynamic Programming" technique.

1. **Maximum sum of a path in a right number triangle**: Given a right triangle of numbers, find the largest of the sum of numbers that appear on a path starting from the top towards the base. The next number on the path is located directly below or below-and-one-place-to-the-right. Show the path discovered.

| Input | Output |
|---|---|
| Content of the *"input_ 1.txt"* file:<br>- $1^{st}$ line: positive integer `n`, represent depth of the triangle.<br>- Next `n` lines: each line contains corresponded positive integers, represent costs of edges, separated by a single space " ". | - Maximum sum of the path.<br>- Cost of the corresponded connected edges, separated by a single space " ". |
| Example:<br>5<br>7<br>8 8<br>4 6 0<br>7 0 8 5<br>0 5 4 9 4 | 38<br>7 8 6 8 9 |

2. **The change-making problem**: Given `k` denominations" $x_1, x_2, ..., x_k$. Find the minimum number of coins (of certain denominations) that add up to a given amount of money `n`. *Note*: Always assume that the smallest coin denomination is 1.

| Input | Output |
|---|---|
| Content of the *"input_ 2.txt"* file:<br>- $1^{st}$ line: `k` positive integers represent `k` denominations, sorted descending, separated by single space " ". The last value must be `1`.<br>- $2^{nd}$ line: Positive integer `n` represents the amount of money required exchange. | - *Denominations 1: amount*<br>- *Denominations 2: amount*<br>- ...<br>- *Denominations k: amount* |
| Example:<br>25 10 5 1<br>72 | 25: 2<br>10: 2<br>5: 0<br>1: 2 |

3. **Longest Common Subsequence (LCS) Problem**: Given 2 strings $S = s_1s_2...s_m$ and $T = t_1t_2...t_n$. find the length of their longest common subsequence and print it.

| Input | Output |
|---|---|
| Content of the *"input_ 3.txt"* file:<br>- String $S_1$<br>- String $S_2$ | - Longest Common Subsequence of $S_1$ and $S_2$. |
| Example:<br>ABCDEFGH<br>DEFGHMNPQ | DEFGH |

4. **Longest Monotonically Increasing Subsequence (LMIS) problem**: Find the length of the longest subsequence of a given sequence of positive integers such that all elements of the subsequence are in monotonically increasing order. Print the longest subsequence.

| Input | Output |
|---|---|
| Content of the *"input_4.txt"* file:<br>- $1^{st}$ line: positive integer `n`, represent size of the array.<br>- $2^{nd}$ line: `n` positive integers, represent elements of the given array. Separated by a single space " ". | - LMIS.<br>- Size of the result array. |
| Example:<br>5<br>4 1 2 6 0 | 1 2 6<br>3 |

5. **Optimal Binary Search Trees**

| Input | Output |
|---|---|
| Content of the *"input_5.txt"* file:<br>- $1^{st}$ line: positive integer `n`, represent number of nodes of optimal BST.<br>- Next `n` lines: each line contains positive integer key $k_i$ and its frequency, separated by a single space " ".<br>- *Note: Sum of frequencies must be 1.* | - Root of the built search tree.<br>- In-order traversal of the built search tree. Elements separated by a single space " ". |
| Example:<br>3<br>1 0.7<br>2 0.2<br>3 0.1 | 1<br>1 2 3 |

6. **Subset-Sum Problem**: Find a subset of a given set $A = a_1.a_2, ..., a_n$ of `n` positive integers whose sum is equal to a given positive integer `k`.

| Input | Output |
|---|---|
| Content of the *"input_6.txt"* file:<br>- $1^{st}$ line: positive integer `n`, represent size of $A$.<br>- $2^{nd}$ line: `n` positive integers represent elements of $A$, separated by a single space " ".<br>- $3^{rd}$ line: positive integer `k`. | - Subsets of $A$ with sum equal to `k` each subsets located on a separate line, elements of each line separated by a single space " ". |
| Example:<br>5<br>7 2 5 4 3<br>9 | 7 2<br>5 4<br>2 3 4 |

7. **Knapsach problem**: Given `n` items of known weights $w_1, w_2, ..., w_n$ and values $v_1, v_2, ..., v_n$ and and a knapsack of capacity $W$. Find the most valuable subset of the items that fit into the knapsack.

| Input | Output |
|---|---|
| Content of the *"input_ 7.txt"* file:<br>- $1^{st}$ line: positive integer $W$ represents capacity of the knapsack.<br>- $2^{nd}$ line: positive integer `n` represents items. These items are numbered from 0 → n - 1<br>- n following lines: $w\_i$ $v\_i$ (represent weight and value of item $i$) | - $1^{st}$ line: ID of the chosen items.<br>- $2^{nd}$ line: Total value of th chosen items |
| Example:<br>20<br>5<br>10 5<br>4 2<br>9 4<br>6 6<br>7 1 | 0 1 3<br>13 |

8. **Traveling Salesman Problem**: The problem asks to find the shortest tour through a given set of `n` cities that visits each city exactly once before returning to the city where it started.

| Input | Output |
|---|---|
| Content of the *"input_ 8.txt"* file:<br>- $1^{st}$ line: positive integer `n` represents `n` cities. Cities are numbered from 1 → n<br>- Following lines: City 1 City 2 Distance<br>- Last line: -1 | - $1^{st}$ line: Traveling order (shortest path).<br>- $2^{nd}$ line: Length of traveled path. |
| Example:<br>5<br>1 2 4<br>3 4 2<br>2 5 1<br>3 5 1<br>-1 | 1 2 5 3 4<br>8 |

## • FILE SUBMISSION REGULATION

- Only submit files with `.cpp` extensions: `1.cpp`, `2.cpp`, ... . **Project submission is illegal**.
- `.cpp` files must be located in `MSSV` folder, then be compressed into `MSSV.zip(.rar)`.
- Source code must receive input and return output as specified for each problem. Submissions with wrong regulation will result in a "0" (zero).
- Plagiarism and Cheating will result in a "0" (zero) for the entire course.
- Contact: **bhthong@fit.hcmus.edu.vn** for more information.

**END**