Lab05-DynamicProgramming

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2020.

$*$ If there is any problem, please contact TA Shuodian Yu.

$*$ Name:Yulong Hui Student ID:518030910059 Email:qinchuanhuiyulong@sjtu.edu.cn

1. **Bookshelf:** Tim has n books and he wants to make a bookshelf to them. The pages' width of the i-th book is w_i and the thickness is t_i.

   Tim puts the books on the bookshelf in the following way. He selects some books and put them vertically. Then the rest of the books are put horizontally above the vertical books. Obviously, the total thickness of the books put vertically must be greater than the sum of widths of the horizontal books. As long as tim wants to make the bookshelf as small as possible, please help him to find the minimum total thickness of the vertical books.

   To simplify the problem, the thickness of each book is either 1 or 2. And all the numbers in this problem are positive integers.

   Design an algorithm based on dynamic programming and implement it in C/C++/Python. The file `Data-P1.txt` is a test case, where the first line contains an integer n. Each of the next n lines contains two integers t_i and w_i denoting two attributes of the i-th book. Source code should be named as *Code-P1.** .You need to briefly describe your algorithm and find the result of `Data-P1.txt` by your program.

   Example:

   Given $n = 5$ books, and $\{(t_i, w_i)|1 \leq i \leq 5\} = \{(1,12),(1,3),(2,15),(2,5),(2,1)\}$. The algorithm should return 5.

   Solution.
   We can deal with this problem by a special dp method.

   First, we define $dp[i][j]$: There are i books in consideration and the total thickness is keeped at j. Then, the value of $dp[i][j]$ means the smallest width of the current books.

   To be more specific, the value means what is the least width we must have to keep the thickness of j. There must be some cases which meet $dp[i][j] > j$. That's to say j thickness is impossible because the width must be smaller than the thickness. Therefore, we can just find the smallest j, which makes $dp[n][j] \leq j$. Then this j is the result.

   Then we just need to calculate the $dp[i][j]$:

   For each (i,j), if $t_i > j$, then we must put that i-th book horizontal, so $dp[i][j] = dp[i-1][j] + w_i$.

   If $t_i \leq j$, then we have to decide which implementation of this book is better, the horizontal one or the vertical one, so $dp[i][j] = min\{dp[i-1][j] + w_i, dp[i-1][j-t_i]\}$

   Then we can write the program, and the result of the example is: 2542.

   □

2. Recall the *String Similarity* problem in class, in which we calculate the edit distance between two strings in a sequence alignment manner.

   You are to find the lowest aligning cost between 2 DNA sequences, in which the cost matrix is defined as

	-	A	T	G	C
-	0	1	2	1	3
A	1	0	1	5	1
T	2	1	0	9	1
G	1	5	9	0	1
C	3	1	1	1	0

where `(-, A)` means adding (or removing) one `A`, etc.

(a) Implement Hirschberg's algorithm with C/C++/Python. Please attach your source code named as Code-P2.*. Your program will be tested against random inputs. Your program should be able to output two sequences after editing.

(b) Using your program, find the edit distance between the two DNA sequences found in attachments `Data-P2a.txt` and `Data-P2b.txt`.

Solution.
(a) The code is attached to the upload zip-file. There are something important I need to explain. In *main*(), there are two methods to get the input (from line260 to line267), you can choose file-stream, or use scanf() function. When switching the two methods, you need to uncomment another one.

(b) The edit distance is 7615. □

由公式 3.22 知；

$$\hat{\beta_1} = \frac{\sum_{i=1}^{n} \hat{r_{i1}} y_i}{\sum_{i=1}^{n} \hat{r_{i1}}^2}$$

$$\hat{\beta_1} = \frac{\sum_{i=1}^{n} \hat{r_{i1}}(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + u_i)}{\sum_{i=1}^{n} \hat{r_{i1}}^2}$$

对于分子：$\sum_{i=1}^{n} \hat{r_{i1}}\beta_0 = 0$
$\sum_{i=1}^{n} \hat{r_{i1}} x_{i2} = 0$
$\sum_{i=1}^{n} \hat{r_{i1}} x_{i1} = \sum_{i=1}^{n} \hat{r_{i1}}^2$

$$\hat{\beta_1} = \beta_1 + \beta_3 \frac{\sum_{i=1}^{n} \hat{r_{i1}} x_{i3}}{\sum_{i=1}^{n} \hat{r_{i1}} * \hat{r_{i1}}} + \frac{\sum_{i=1}^{n} \hat{r_{i1}} u_i}{\sum_{i=1}^{n} \hat{r_{i1}} * \hat{r_{i1}}}$$

$$E(\hat{\beta_1}) == \beta_1 + \beta_3 \frac{\sum_{i=1}^{n} \hat{r_{i1}} x_{i3}}{\sum_{i=1}^{n} \hat{r_{i1}} * \hat{r_{i1}}}$$

Remark: You need to include your .pdf and .tex and 2 source code files in your uploaded .rar or .zip file.