CIS 315

June 8

ACM Prog Contest

CIS 407 (Fall 2018) \leq \text{Plaza active now}
2. Recall the midterm problem: imagine we are on a tour of some popular tourist spots and we wish to visit each one of them. They are all along a single road at mileposts \( m_0, m_1, m_2, \ldots, m_n \), and we will stop at each one in that order. The choice we have to make is which taxi company to use to take us from one location to the next.

The choices are taxi company \( S \) (slow) and company \( T \) (tedious). Company \( S \) charges \( s_i \) dollars per mile to travel from location \( i \) to \( i + 1 \), so the charge for that segment is \((m_{i+1} - m_i) \cdot s_i\) dollars. Company \( T \) charges a flat rate of \( t \) dollars per segment but if chosen they must be used for three consecutive segments. For example, we could start with company \( T \) at location \( i \), have them take us to \( i + 1 \), \( i + 2 \), and finally finish with them at location \( i + 3 \) for a total of \( 3t \) dollars (of course, we could decide to use them for the next three segments). Our goal is to find the cheapest cost for all segments taking us from location \( 0 \) to location \( n \).

For example, suppose the mileposts are at locations \((0, 25, 50, 100, 150, 200)\), the costs for company \( S \) are \((2, 3, 2, 1, 2)\), and the flat rate for company \( T \) is 40. If the travel plan is \((S, T, T, T, S)\), the travel cost is \(25 \cdot 2 + 40 + 40 + 40 + 50 \cdot 2 = 270\) dollars. However the plan \((S, S, T, T, T)\) incurs a cost of \(25 \cdot 2 + 25 \cdot 3 + 40 + 40 + 40 = 245\) dollars.

Define the subproblem \( C(i) \) to be the minimum cost of a plan that starts at location \( 0 \) and ends at location \( i \), and so that either \( (a) \) company \( S \) brought us from \( m_{i-1} \) to \( m_i \) (at a cost of \( s_{i-1} \) per mile, or \( b) \) company \( T \) was used on the last three segments (from \( m_{i-3} \) to \( m_{i-2} \) to \( m_{i-1} \) to \( m_i \)). A recurrence for \( C \) is given by

\[
C(i) = \begin{cases} 
0 & \text{if } i = 0 \\
(m_i - m_{i-1}) \cdot s_{i-1} + C(i - 1) & \text{if } 0 < i \leq 2 \text{ (can only use company } S) \\
\min\{(m_i - m_{i-1}) \cdot s_{i-1} + C(i - 1), 3 \cdot t + C(i - 3)\} & \text{if } i \geq 3 \text{ (min of choices (a) and (b) above)}
\end{cases}
\]

The minimum cost to go from location \( 0 \) (at \( m_0 \)) to location \( n \) (at \( m_n \)) You are to write pseudo-code which will fill up an array \( C \) in either a bottom-up (iterative) manner or in a top-down (memoized) manner (you may use a hash table in the memoized case if you wish).

**PART I:** My pseudo-code is **ITERATIVE** or **MEMOIZED** (circle one)

**PART II:** The time bound of my pseudo-code is:

\[ \Theta(n) \]

(continue on next page)
PART III: Write your pseudo-code here.

create array C of int
C[0] = 0

for i = 1 to 2
    C[i] = (m_i - m_{i-1}) \cdot S_{i-1} + (L_{i-1})

for i = 3 to n
    \[
    C[i] = \min \left( (m_i - m_{i-1}) \cdot S_{i-1} + C[i-1], 3t + C[i-3] \right)
    \]

if 1st value then min
    P[i] = "S"
else
    P[i] = "T"

true \( O(n) \)

reconstruction phase

\( i = n \)

while \( i > 0 \)
    if \( P[i] = S \)
        then print use \( S \) from \( i-1 \) to \( i \)
        \( i = i-1 \)
    else
        print use \( T \) from \( i-3 \) to \( i \)
        \( i = \frac{i}{4} \) i-3
4. We have the problem of deciding whether two strings $X$ and $Y$ can be inter-leaved (or shuffled) in a manner that forms a given third string $Z$. As part of this process, you need to devise a recurrence relation that could be used to devise a dynamic programming algorithm. The specifics are

- the three strings are $X = x_1x_2\ldots x_n$, $Y = y_1y_2\ldots y_m$, and $Z = z_1z_2\ldots z_{n+m}$
- all characters of $X$ must appear in $Z$ in the same order (but not necessarily consecutively)
  - in other words, $X$ is a subsequence of $Z$
- $Y$ is a subsequence of $Z$
- all characters in $Z$ match a character of either $X$ or $Y$

For example, suppose $X = aacb$, $Y = bcaba$, and $Z = abacabcba$. Then the answer is True, since we can write $Z = AbAcAbCBA$, where the upper-case represents locations of $X$ and lower-case $Y$.

Define a subproblem $L(i, j)$ to be True if $x_1x_2\ldots x_i$ and $y_1y_2\ldots y_j$ can be inter-Leaved to form $z_1z_2\ldots z_{i+j}$, and False otherwise.

Give a recurrence for the function $L$. Be sure to include base and boundary cases.

\[
L(i, j) = \begin{cases} 
\text{True} & i = j = 0 \\
\quad \frac{x_i = z_i \land L(i-1, 0)}{x_i = z_i} & i > 0 \land j = 0 \\
\quad \frac{y_j = z_j \land L(0, j-1)}{y_j = z_j} & i = 0 \land j > 0 \\
\quad \frac{(x_i = z_{i+j} \land L(i-1, j)) \quad \text{or} \quad (y_j = z_{i+j} \land L(i, j-1))}{(x_i = z_{i+j}) \land L(i-1, j)} & i > 0, j > 0
\end{cases}
\]

\[\land = \text{AND}\]

Return $L(n, m)$

\[O(nm) \text{ space}\]

For $i = 1$ to $n$
For $j = 1$ to $m$