CSC373 Winter 2015 Problem Set # 4

Name: Weidong An
Student Number: 1000385095
UTOR email: weidong.an@mail.utoronto.ca
February 2, 2015

(a) **Recursive Structure**
The minimum total time required MinimumTime($s_1 ... s_m$) can be written as
MinimumTime($s_1 ... s_{p_c}$) + MinimumTime($s_{p_c+1} ... s_m$) + $|m|$
recursively for some $1 \leq c \leq k$ and q_c is the first break point.

(b) **Array Definition**
For convenience, let $q_0 = 0$ and $q_{k+1} = m$.
Define array $T[i, j]$ with $0 \leq i < j \leq k + 1$.
$T[i, j]$ = minimum time required to break the string $s_{p_i+1} ... s_{p_j}$.

(c) **Recurrence Relation**
Base case: $T[i, i + 1] = 0$
General case($j - i > 1$): $T[i, j] = \min(T[i, b] + T[b, j]) + |p_j - p_i|$ for all $i + 1 \leq p \leq j - 1$.
Explanation: For $T[i, j]$, if $j = i + 1$, then there is no breakpoint, so 0 is as desired. Otherwise $j - i > 1$, there must be a break point p_b that is chosen first to minimize the total time. Then the time required is the sum of time of breaking $s_i ... s_{p_b}$ and $s_{p_b+1} ... s_j$ plus $|p_j - p_i|$.

(d) **Iterative Algorithm**
The following pseudocode is based on the recurrence relation defined above.

1 **for** $t = 1, ..., k + 1$
2   **for** $i = 0, ..., k - t + 1$ # i denotes the row, $i + t$ denotes the column
3   **if** $t == 1$
4             $T[i, i + t] = 0$
5   **else**
6             $T[i, i + t] = \infty$
7   **for** $p = i + 1, ..., i + t - 1$ #This loop is to find the minimum
8   **if** $T[i, p] + T[p, i + t] + |p_{i+t} - p_i| < T[i, i + t]$
9                     $T[i, i + t] = T[i, p] + T[p, i + t] + |p_{i+t} - p_i|$

(e) **Reconstruct Solution**
Define another array B to store the result. $B[i, j] = b$ means the first break point in $s_{p_i+1} ... s_{p_j}$ chosen is p_b. The following pseudocode is the same as above except line 9 and line 11.

```
for t = 1, ..., k + 1
    for i = 0, ..., k − t + 1
        if t == 1
            T[i, i + t] = 0
        else
            T[i, i + t] = ∞
            for b = i + 1, ..., i + t − 1
                if T[i, b] + T[b, i + t] + |p_{i+t} − p_i| < T[i, i + t]
                    intermed = b
                    T[i, i + t] = T[i, b] + T[b, i + t] + |p_{i+t} − p_i|
            B[i, i + t] = intermed
```

To get the optimum permutation, first define the recursive function.

```
GET-OPTIMAL-PERMUTATION(B, i, j)
if j − i == 1
    return []
else
    return [B[i, j]] + GET-OPTIMAL-PERMUTATION(B, i, B[i, j])+
    GET-OPTIMAL-PERMUTATION(B, B[i, j], j)
```

Then implement the following:

```
return GET-OPTIMAL-PERMUTATION(B, 0, k + 1)
```

Because we have set $p_0 = 0$ and $p_{k+1} = m$, the result is in terms of $s_1 ... s_m$.