Master 1 MoSIG

Algorithmic Problem Solving

APP3 Report
Hole Drilling
Team:
SACAD

Members:
Andrey **SOSNIN**
Majdeddine **ALHAFEZ**
Antoine **COLOMBIER**
Eman **AL-SHAOUR**
Son Tung **DO**

Grenoble, 12 November, 2017

1 Introduction

We want optimal. Naive approach - exponential, we use dynamic instead and yata-yata-yata. D is the dictionary with latin characters, $\mathcal{D}$ is its translation into Morse. We suppose that $\epsilon \notin D$.

2 Principle

We denote the Morse sequence by $m = (m_0 \dots m_{L-1})$. $m[i,j]$ denotes the subsequence $(m_i \dots m_{j-1})$.
The proposed dynamic procedure is based on the following observation:
Let w be a word of the dictionary such that its unique translation into Morse alphabet w' of length $|w'|$ occurs in m at index $i > 0$ (i.e. $m[i, i + |w'|] = w'$). Then the number $C_{i,w'}$ of interpretations of $m[0, i + |w'|]$ where the last word is w, equals the number C_i of interpretations of $m[0,i]$.
This leads us to establish a recursive formula for C_i:

$$C_i = \sum_{\substack{w' \in \mathcal{D} \\ m[0,i] = \alpha w'}} C_{i-|w'|,w'} + F_i = \sum_{\substack{w' \in \mathcal{D} \\ m[0,i] = \alpha w'}} C_{i-|w'|} + F_i$$

where $C_0 = 0$, "$m[0,i] = \alpha w'$" means that w' is a strict suffix of $m[0,i]$ and

$$F_i = |\{w' \in \mathcal{D} \mid m[0,i] = w'\}| \quad (note\ that\ F_i = 0\ for\ i > 4 \times M)$$

3 Algorithm

---

procedure *compute_number_sequences*$(m, L, \mathcal{D})$
    $C \leftarrow$ *create_array*$(L, 0)$;
    for $i = 1 \dots L-1$ **do**
        for $w' \in \mathcal{D}$ **do**
            if $m[i - |w'|,\ i] = w'$ **then** ▷ the word matches the sequence
                if $|w'| = i$ **then** ▷ is it the first word in m?
                    $C[i] = 1$
                else
                    $C[i] += C[i - |w'|]$
                end if
            end if
        end for
    end for
    return $C[L-1]$
end procedure

---

4 Complexity analysis

Time complexity

At every iteration inside the innermost loop, aside from some constant-cost operations, we perform an equality check between two strings of length bound by M: it costs $O(M)$. In total there are $L \times N$ of those iterations: the total procedure's time cost is in $O(L \times N \times M)$. It is polynomial in all variables.

Space complexity

We have to produce an array of size L. Also, if we consider that the dictionary $\mathcal{D}$ of words translated into Morse is not available at the beginning, storing it would cost an additional $O(N \times M)$ overhead – however, iterating on words of D (rather than $\mathcal{D}$) and translating them on-the-fly would neither increase the asymptotic space complexity nor the time complexity (as it would take $O(M)$ operations each time and would count additively).

5 Improvements

The first improvement to the space complexity would consist in storing at any given point only last 80 computed values of C, as there can be no words longer than $4 \times M = 80$ Morse symbols. For that we should replace $C[x]$ by $C[x\ mod\ 4 \times M]$ for all x (as well as $create\_array(L, 0)$ by $create\_array(4 \times M, 0)$)