

BruteForce vs Dynamic

Max Maldonado
Universidad de Artes Digitales

Guadalajara, Jalisco

Email: idv16a.msolano@uartesdidgitales.edu.mx

Profesor: Efraín Padilla

Agosto 10, 2019

I. DEFINICIÓN DEL PROBLEMA

En este ejercicio ilustramos las ventajas de utilizar algoritmos dinámicos sobre los algoritmos "dump" o "brute force". El problema reside en la necesidad de mejorar la velocidad de los algoritmos recursivos.

II. INPUTS Y OUTPUTS

Input: Par de Cadenas de SubSecuencias
Output: Subsecuencia común más largar

III. SOLUCIÓN

A. Fuerza Bruta

El algoritmo por fuerza bruta pretende realizar búsquedas exhaustivas entre todas las posibles combinaciones, sin utilizar ningún tipo de "memoria" que le permite realizar revisiones más inteligentes. El algoritmo no utiliza mucha memoria, pero es lento.

B. Dinámico

El algoritmo dinámico tiene la ventaja de utilizar un sistema de "memoria" que le permite revisar subsecuencias sin realizar precalculos. Aunque utiliza un poco más de memoria, la rapidez es mucho mayor, pasando de minutos (brute forces) a milisegundos.

IV. CODE

A. Brute Force

```

vector<string>
getSubsequences
(
    const string & _subsequence,
    const unsigned int _number
)
{

    unsigned int string_length = _subsequence.length();
    int subsequence_length = pow(2, string_length);

    vector<string> subsequence_list;
    for (unsigned int counter = 1; counter < subsequence_length; ++counter)
    {
        string subsequence = "";
        for (unsigned int j = 0; j < string_length; ++j)
        {
            if (counter & (1 << j))
            {
                subsequence.push_back(_subsequence[j]);
            }
        }

        subsequence_list.push_back(subsequence);
    }

    return subsequence_list;
}

string
subsequenceComparison
(
    const vector<string> & _sub_1,
    const vector<string> & _sub_2
)
{
    string lcs = "";

    unsigned int sub_1_len = _sub_1.size();
    unsigned int sub_2_len = _sub_2.size();

    std::cout << " len 1 " << sub_1_len;
    std::cout << " len 2 " << sub_2_len;

    for (unsigned int i = 0; i < sub_1_len; ++i)
    {
        for (unsigned int j = 0; j < sub_2_len; ++j)
        {
            if (_sub_1[i] == _sub_2[j])
            {
                if (lcs.size() < _sub_1[i].size())
                {
                    lcs = _sub_1[i];
                }
            }
        }
    }
}

```

```
    }  
}  
  
    std::cout << "it: " << i << "\n";  
}  
  
return lcs;  
}
```

B. Dynamic

```

string
LCS
(
    const string & _sequence_1,
    const string & _sequence_2
)
{
    const size_t m = 15;
    const size_t n = 16;

    int matrix[m+1][n+1] {};

    for (unsigned int i = 0; i < m + 1; ++i)
    {
        for (unsigned int j = 0; j < n + 1; ++j)
        {
            if (i == 0 || j == 0)
            {
                matrix[i][j] = 0;
            }
            else if (_sequence_1[i - 1] == _sequence_2[j - 1])
            {
                matrix[i][j] = matrix[i - 1][j - 1] + 1;
            }
            else
            {
                matrix[i][j] = std::max(matrix[i - 1][j], matrix[i][j - 1]);
            }
        }
    }

    int index = matrix[m][n];

    string lcs;
    lcs.resize(index + 1, '#');
    lcs[index];

    int i = m;
    int j = n;

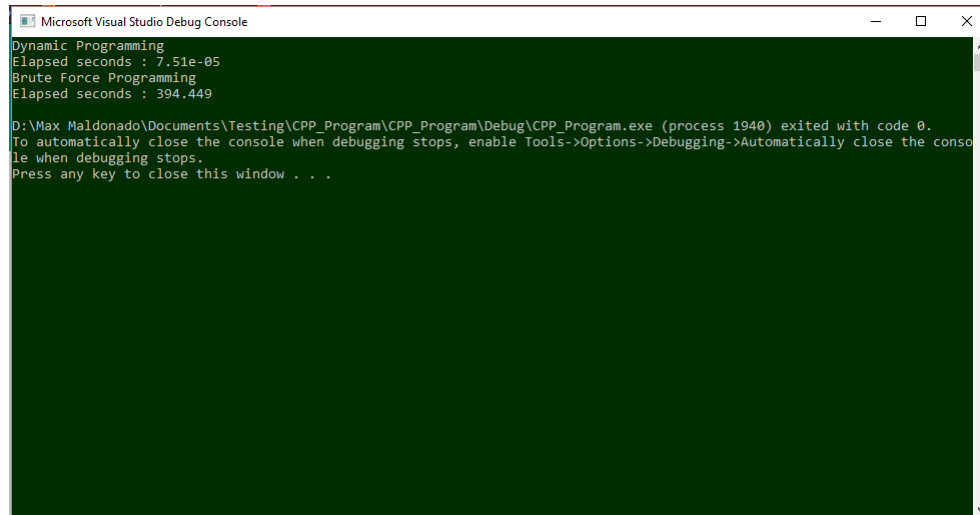
    while (i > 0 && j > 0)
    {
        if (_sequence_1[i - 1] == _sequence_2[j - 1])
        {
            lcs[index - 1] = _sequence_1[i - 1];

            —i;
            —j;
            —index;
        }
        else if (matrix[i - 1][j] > matrix[i][j - 1])
        {
            —i;
        }
        else
    }

```

```
        {  
            —j;  
        }  
    }  
    return lcs;  
}
```

V. COMPARACIÓN



```
Microsoft Visual Studio Debug Console
Dynamic Programming
Elapsed seconds : 7.51e-05
Brute Force Programming
Elapsed seconds : 394.449

D:\Max Maldonado\Documents\Testing\CPP_Program\CPP_Program\Debug\CPP_Program.exe (process 1940) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```