Jeffrey Liv

[Jeffro001@csu.fullerton.edu](mailto:Jeffro001@csu.fullerton.edu)

Alex Athas

[genesiscloud@csu.fullerton.edu](mailto:genesiscloud@csu.fullerton.edu)

Project 3 – String Matching with Dynamic Programming – Part 1

Find\_max and dynamic\_lcs functions:

Analysis:

Find\_max takes constant time

|  |
| --- |
| int find\_max(int num1, int num2, int num3) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| int max = num1; |
|  |

|  |
| --- |
| if (num2 > max) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| max = num2; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| if (num3 > max) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| max = num3; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return max; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| // ------------------------------------------------------------------------- |
|  |

|  |
| --- |
| int dynamicprogramming\_longest\_common\_subsequence(const std::string & string1, |
|  |

|  |
| --- |
| const std::string & string2) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| const int n = string1.size(); |
|  |

|  |
| --- |
| const int m = string2.size(); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int max\_len = 0; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (n == 0 && m == 0) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| return 0; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| if (n == 1 && m == 1) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| return 1; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int up, left, diag; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int D[n + 1][m + 1]; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| for (int i = 0; i < n; i++) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| D[i][0] = 0; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| for (int j = 0; j < m; j++) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| D[0][j] = 0; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| for (int i = 1; i < n; i++) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| for (int j = 1; j < m; j++) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| up = D[i - 1][j]; |
|  |

|  |
| --- |
| left = D[i][j - 1]; |
|  |

|  |
| --- |
| diag = D[i - 1][j - 1]; |
|  |

|  |
| --- |
| if (string1[i - 1] == string2[j - 1]) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| diag = diag + 1; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| D[i][j] = find\_max(up, left, diag); |
|  |

|  |
| --- |
| max\_len = D[i][j] + 1; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| return max\_len; |
|  |

}