

Universidad Simón Bolívar

Departamento de Computación y Tecnología de la Información

CI 5651 - Diseño de Algoritmos

**Proyecto 3 – Programacion Dinamica**

Grupo:

David Lilue 09-10444

**1. ACMAKER**

La idea principal en la solucion de este problema es recorrer letra por letra la abreviacion. Separando la oracion en palabras. Se verifica si la letra actual se encuentra en la palabra actual de la oracion. Si es asi, queda marcada la posicion de la misma, para luego comprobar si la siguiente letra esta en esa misma palabra, sino sigue a la proxima palabra. Aunque, si se encuentran todas la letras antes de que se acaben las palabras, el caso no genera una posible abreviacion.

La estrategia que se penso usar, fue una matriz de dos dimensiones, pero no logre que guardara los valores, en la recursiones repetidas. Usando programacion dinamica, top-down.

La complejidad es de O(n\*m), siendo n el tamño de la abreviacion y m la cantidad de palabras en la oracion.

4. MAXWOODS

La solucion que se ideo fue, partiendo de la posicion (0,0) de la matriz de los arboles, probar moverse horizontalmente y hacia abajo. Tomando el maximo numero de arboles coortados de ambas opciones y sumando si en la posicion actual habia un arbol. La direccion del hombre se inferia por la posicion en la fila, invirtiendola en cada llamada recursiva que bajaba en las filas.

Una estrategia que se pensaba usar, era tener una matriz de dos dimensiones pero no se podia. Y no supe como hacerlo con una dimension. Supongo que era tomando en cuante las columnas, es decir, si era bottom-up e iba desde la posicion (m-1,n-1). Recorria la fila primero y por filas solo necesitaba saber un acumulado en cada columna y tomando en cuenta el maximo por esa fila en direccion adecuada.