

Juan Camilo Buitrago

CHALLENGE: DYNAMIC PROGRAMMING

Algoritmo: El programa recibe como parámetro un archivo txt que contiene en la primera fila la cantidad de cuartos que contiene la lista, en la primera columna el nombre de los cuartos y en la segunda columna la cantidad de basura que contienen. Sin embargo, para el programa solo va a utilizar la segunda columna (la cual llamaremos “basura”) para encontrar la solución.

Una vez recibe por parámetro a la lista “basura”, genera un diccionario de arreglos sin longitud definida, pero con la misma cantidad de arreglos que posiciones en “basura”. En caso que “basura” tenga 20 elementos, se va a generar un diccionario que contiene 20 listas vacías. Seguido, se agrega el primer valor del arreglo de basura a la primera lista y empieza el proceso de encontrar el camino más largo. Para esto se van a hacer 2 recorridos, uno sobre todo el arreglo (i) y otro sobre los valores anteriores al valor actual (j). Por ende, el recorrido tiene que empezar por la posición 1 (i) para que la primera comparación sea con la posición 0 (j). Después, la referencia será la posición 2 (i) y se comprará con las posiciones 0 y 1 (j ambos). El primer filtro es si el valor en j es mayor a la referencia (i), si esto se cumple, revisa si la lista en j es más grande que en i, de ser verdadero, se copia la lista de j y se le agrega el valor de i. Este proceso asegura que se esté encontrando el camino más largo. De no ser verdadera alguna de las comparaciones, solamente se agrega el valor i a la lista i, esto también asegura que se estén guardando todos los caminos posibles. Paralelo a esto, hay otro arreglo que guarda las posiciones de todos los caminos generados.

La función va a retornar un diccionario lleno de caminos y finalmente se hace un recorrido para hallar el camino más largo y se utiliza la lista de posiciones para recuperar los valores tanto de basura como los nombres de la lista original. Por último, se genera un archivo txt con esta respuesta, en el mismo formato que el archivo ingresado.