uniandes

Ingeniería de Sistemas y Computación

Diseño y análisis de algoritmos

Profesor: Mateo Sanabria Ardila

Taller programación dinámica

Fecha de entrega: 11/Marzo | 2023-1 | Nota máxima: 50

Solucione los problemas propuestos usando el lenguaje de su preferencia (`python,java,c`) de ser posible entregue todas las soluciones en un mismo archivo, podría crear una clase `solution` y ahí definir los métodos propuestos. Si toma esta opción, su clase y métodos **NO** pueden tener otro nombre. La otra opción es que solucione los problemas usando funciones, este caso también tenga en cuenta que el nombre de las funciones no puede ser diferente al propuesto. Para cada uno de los siguientes problemas proponga una implementación usando programación dinámica que cumpla con la complejidad temporal requerida, puede usar comentarios para justificar la complejidad temporal de su explicación o si lo cree conveniente puede agregar un archivo `PDF` en el que explique esto para cada problema.

1. **Subsecuencia creciente de suma máxima** Este problema consiste en encontrar la subsecuencia creciente más larga de un arreglo de enteros que tenga suma máxima, para este problema se requiere la suma de dicha subsecuencia y no la subsecuencia como tal. Implemente la función `maximum_sum_increasing_subsequence`, con complejidad temporal $\mathcal{O}(n^2)$.

2. **Subsecuencia palindrome más larga** Este problema consiste en encontrar la subsecuencia más largas de una cadena que sea palíndromo. Implemente la función `longest_palindromic_subsequence`, con complejidad temporal $\mathcal{O}(n^2)$.

3. **Subcadena repetida más larga** Dados dos cadenas de caracteres este problema consiste en encontrar la subcadenda mas larga que esta contenida en las dos cadenas dadas. Implemente la función `longest_common_substring`, con complejidad temporal $\mathcal{O}(n \times m)$.