

Programación III
Práctica Final
Programación Dinámica

1. **Nombre y apellido de los miembros del grupo**

Juan José Bello Santana
Aarón Hernández Álvarez

2. **Dirección de correo de los miembros del grupo**

juan.bello103@alu.ulpgc.es
aaron.hernandez104@alu.ulpgc.es

3. **Enunciado del problema**

El objetivo del problema es retornar la subsecuencia palindrómica más larga dentro de una cadena de texto ofrecida.

4. **Referencias (enlace a páginas web utilizadas)**

<https://www.techiedelight.com/longest-palindromic-subsequence-using-dynamic-programming/>
<http://techieme.in/longest-palindromic-subsequence/>
<https://www.geeksforgeeks.org/longest-palindromic-subsequence-dp-12/>

5. **Recurrencia**

$$LPS(i..j) \begin{cases} 1 \rightarrow (if\ i = j) \\ LPS[i + 1..j - 1] + 2 \rightarrow (if\ X[i] = X[j]) \\ \max(LPS[i + 1..j], LPS[i..j - 1]) \rightarrow (if\ X[i] \neq X[j]) \end{cases}$$

6. **Explicación de la recurrencia**

La búsqueda del palíndromo más largo se basa en la llamada recursiva modificando los punteros i y j conforme se suceden los casos expresados arriba.

- Si los punteros son iguales, entonces es que es palíndromo y retornamos 1.
- Si no fuese así, pero el valor de la celda de texto en sendos punteros fuera el mismo, se da la situación de que existe un posible palíndromo, por lo cual retornamos 2 (la longitud del palíndromo actual) más el resultado de las sucesivas llamadas recursivas.
- Ante otros casos en los que el valor de la cadena en ambos punteros es diferente, tenemos que se realiza la expansión del árbol de llamadas decrementando j o incrementando i y prosiguiendo con el árbol restante de llamadas

7. **Análisis asintótico (memoization/tabulation)**

MEMOIZATION

FASE 1

La ecuación de recurrencia empleada en ambos casos es muy similar a otras ya vistas en clase.

En el caso de la técnica de memoization, se puede observar que las dos primeras operaciones constituyen casos de orden 1 para la igualdad entre punteros y de orden n para la llamada recursiva única que incrementa y decrementa, respectivamente, los punteros i y j .

Si nos situamos en el peor de los casos, aquel en el que las dos letras sean diferentes, nos encontramos con dos llamadas recursivas hacia el mismo método, lo cual se trata pues de una operación de $O(2^n)$. Sin embargo, si tenemos en cuenta que estamos mapeando las soluciones de aquellos subproblemas ya resueltos, obtenemos un ahorro en el costo operacional importante, por lo que la reducción del orden de operación lo reduce a $O(n^2)$

Si aplicamos la regla de la suma, tendremos que $O(1) + O(n) + O(n^2)$ resulta en $O(n^2)$ para la fase 1

FASE 2

La estructura lógica de recorrido del mapa usa los mismos pasos, por lo que tenemos exactamente el mismo orden de complejidad que se muestra en el caso anterior, es decir $O(n^2)$, ya que el peor caso es una sucesión de llamadas recursivas constantes en las cuales nunca se forma ningún palíndromo y aquellas repetidas ya resueltas se almacenan en el diccionario.

TABULATION

FASE 1

En esta primera fase, en la que se calcula la longitud del palíndromo más largo, se haya un doble bucle for en donde ambos recorren todos los elementos de la matriz (desde 1 hasta n) y el resto de operaciones son de asignación de valores a posiciones de la matriz ($O(1)$), por lo tanto $n * n = n^2$.

FASE 2

La estructura lógica del recorrido de la matriz es similar al anterior, por lo que la complejidad también es de $O(n^2)$. Esto se debe a que en el caso de no formarse ningún palíndromo, nunca se recorrería la diagonal de la matriz, por lo que se tendría que recorrer la matriz $n * n$ veces, es decir $O(n^2)$

8. Copia de pantalla que muestre el uso del programa Python desde consola activando combinaciones de todos los switches obligatorios.

python Main.py -f ./files/data_1 -t -sm

```
(base) PS D:\Universidad\OneDrive - Universidad de Las Palmas
de Gran Canaria\TERCERO\Programacion III\Entrega Final P2>
python Main.py -f ./files/data_1 -t -sm

*****
./files/data_1: ESTRATEGIA SELECCIONADA: MEMORIZALIZACIÓN
Para el candidato: ieuyrlalsdjdslaluroqwue
Se encuentran los siguientes resultados:
Maxima Longitud: 17
Palindromo resultante: eurlalsdjdslalrue
Tiempo de ejecución completo (µs): 1995
```

python Main.py -d ./files -t -st --check

```
(base) PS D:\Universidad\OneDrive - Universidad de Las Palmas
de Gran Canaria\TERCERO\Programacion III\Entrega Final P2>
python Main.py -d ./files -t -st --check

*****
./files/data_1: ESTRATEGIA SELECCIONADA: TABULACIÓN - ST
Para el candidato: ieuyrlalsdjdslaluroqwue
Se encuentran los siguientes resultados:
Maxima Longitud: 17
Palindromo resultante: eurlalsdjdslalrue
RESOLUCIÓN POR METODO ALTERNATIVO (memoization) -- CHECK
Maxima Longitud: 17
Palindromo resultante: eurlalsdjdslalrue
Tiempo de ejecución completo (µs): 1969

*****
./files/data_2: ESTRATEGIA SELECCIONADA: TABULACIÓN
Para el candidato: babcbabcb
Se encuentran los siguientes resultados:
Maxima Longitud: 7
Palindromo resultante: bcbabcb
RESOLUCIÓN POR METODO ALTERNATIVO (memoization)
Maxima Longitud: 7
Palindromo resultante: babcbab
Tiempo de ejecución completo (µs): 1994
```

9. Copia de pantalla que muestre el uso del programa Java desde consola activando combinaciones de todos los switches obligatorios.

```
java -jar LPSJava.jar -d C:\Users\byjua\OneDrive\Escritorio\TerceroEII\IS2\LPSJava\files -t -st --check
```

```
PS C:\Users\byjua\OneDrive\Escritorio\TerceroEII\IS2\LPSJava\dist> java -jar LPSJava.jar -f C:\Users\byjua\OneDrive\Escritorio\TerceroEII\IS2\LPSJava\files\data_1 -t -sm
Para el candidato: ieuyrlalsdjdslaluroqwue
Se encuentran los siguientes resultados:

Maxima Longitud: 17
Palindromo resultante: eurlalsdjdslalrue
Tiempo de ejecución completo (microsegundos): 1.61055181E9
```

```
java -jar LPSJava.jar -f
C:\Users\byjua\OneDrive\Escritorio\TerceroEII\IS2\LPSJava\files\data_1 -t -sm
```

```
PS C:\Users\byjua\OneDrive\Escritorio\TerceroEII\IS2\LPSJava\dist> java -jar LPSJava.jar -d C:\Users\byjua\OneDrive\Escritorio\TerceroEII\IS2\LPSJava\files -t -st --check
=====
ARCHIVO: data_1
=====
Para el candidato: ieuyrlalsdjdslaluroqwue
Se encuentran los siguientes resultados:

Maxima Longitud: 17
Palindromo resultante: eurlalsdjdslalrue

Verificación con Memoization
Máximo: 17
Palíndromo escogido: eurlalsdjdslalrue
Tiempo de ejecución completo (microsegundos): 0.009

=====
ARCHIVO: data_2
=====
Para el candidato: babcbabcb
Se encuentran los siguientes resultados:

Maxima Longitud: 7
Palindromo resultante: bcbabcb

Verificación con Memoization
Máximo: 7
Palíndromo escogido: babcbab
Tiempo de ejecución completo (microsegundos): 0.003

=====
ARCHIVO: data_3
=====
Para el candidato: bacb
Se encuentran los siguientes resultados:

Maxima Longitud: 3
Palindromo resultante: bcb

Verificación con Memoization
Máximo: 3
Palíndromo escogido: bab
Tiempo de ejecución completo (microsegundos): 0.015

=====
ARCHIVO: data_4
=====
Para el candidato: abbca
Se encuentran los siguientes resultados:

Maxima Longitud: 4
Palindromo resultante: abba

Verificación con Memoization
Máximo: 4
Palíndromo escogido: abba
Tiempo de ejecución completo (microsegundos): 0.016

=====
```

10 . Añade aquí cualquier comentario o aclaración que consideres necesaria sobre tu ejercicio

Nada que reseñar