DOCUMENTO TÉCNICO

# Requisitos

## Especificación

#RESUMEN DEL PROBLEMA (opcional)

Buscar el modo de invertir una lista de números

### Entrada:

### Una lista de n elementos

### Salida:

### La lista de n elementos de manera invertida

# Diseño

## Estrategia

La solución de este problema es parecida a la de organizar la lista tomando el menor de los elementos. En este caso, se toma el ultimo elemento de la lista y se pega la función que toma ese ultimo elemento de la lista, pero esta lista ya no va a ser la original, sino va a ser la lista sin contar el ultimo elemento de la lista del inicio. Este proceso se repetirá hasta que no queden mas elementos en la lista, así quedando la lista organizada

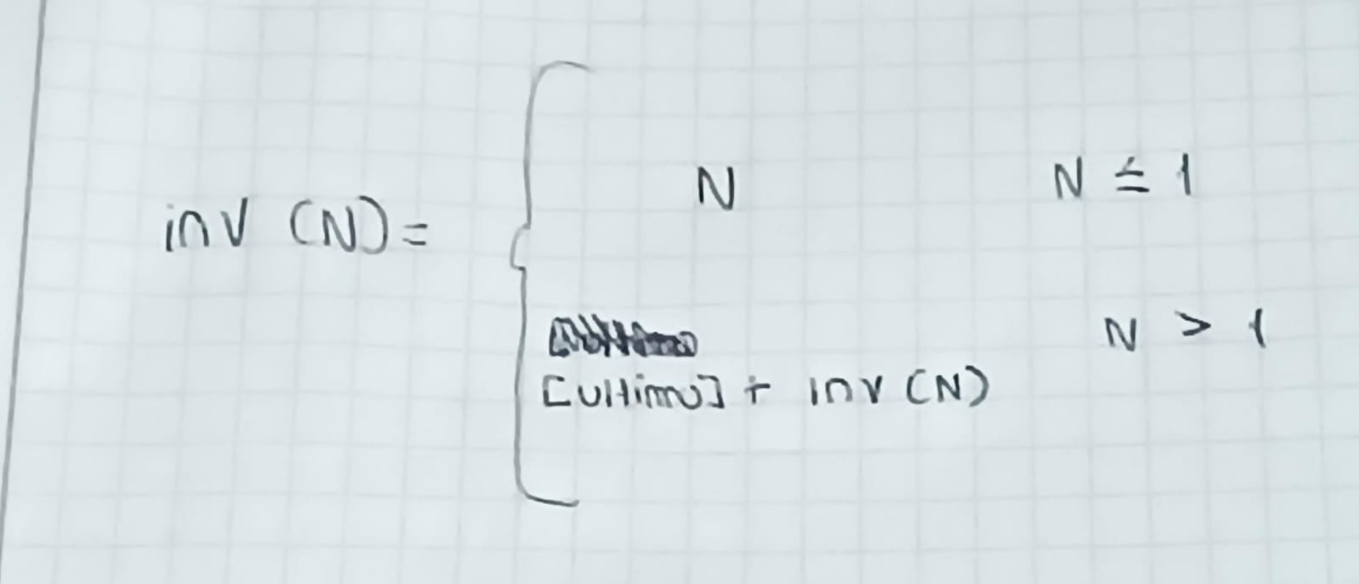
En la función a trozos se tienen que considerar 2 casos

1. Si la longitud de n es menor o igual que 1 entonces la respuesta será n ya que no hay elementos para invertir
2. Si la longitud de n es mayor que 1 entonces va a tomar el último elemento de la lista y le va a agregar el “penúltimo” elemento de la lista sin contar el último elemento y así sucesivamente.

#Estructuras de datos

### Se usaron las listas para almacenar los datos invertidos

#ALGORITMO



## Casos de prueba

|  |  |  |
| --- | --- | --- |
| **Entrada** | **Justificación** | **Salida** |
|  |  |  |
| Inv ([1]) | caso base, lista de tamaño 1 | 1 |
| Inv ([1,2,3,4,5]) | Caso común | [5,4,3,2,1] |
|  |  |  |

# Análisis

## Temporal

Este algoritmo es demasiado similar a insertion sort, ya que en este caso coge el ultimo numero de la lista y le pega el resto (proceso prácticamente idéntico que se realizaba en insertion sort); por esto el costo será de n.

# Código

*Al igual que el punto anterior pueden guiarse con este ejemplo*

## Documentación

Dentro del código.

## Fuentes

*Las fuentes son el código sobre el cual escribieron el documento*

/arena\_1\_a