

Complejidad Computacional

Nombre: Dennys Alexander Pucha Carrera

Fecha: 17-08-2023

Tema: Lección Unidad 3

Resolución:

1. Diseñar un algoritmo voraz (también llamados ávidos o glotones).

Algoritmo de ordenamiento de diferencia absoluta: Es un algoritmo que funciona de tal forma que organiza una lista de números de manera especial. Su objetivo es que los números en la lista estén en un orden que minimice la diferencia entre los números vecinos.

Pseudocódigo:

```
1  Algoritmo ordenamiento_diferencia_absoluta(lista):
2      Ordenar(lista)
3
4      lista_tamano = Tamaño(lista)
5      resultado = CrearArreglo(lista_tamano)
6
7      mitad = lista_tamano // 2
8      resultado[0] = lista[mitad]
9
10     derecha = mitad - 1
11     izquierda = mitad + 1
12     i = 1
13     Mientras i < lista_tamano Hacer:
14         Si i MOD 2 == 1 Entonces
15             resultado[i] = lista[derecha]
16             derecha = derecha - 1
17         Sino
18             resultado[i] = lista[izquierda]
19             izquierda = izquierda + 1
20         Fin Si
21         i = i + 1
22     Fin Mientras
23
24     Devolver resultado
25 Fin Algoritmo
```

Código implementado en Python:

```
1 def ordenamiento_diferencia_absoluta(lista):
2     lista.sort()
3
4     lista_tamano = len(lista)
5     resultado = [0] * lista_tamano
6
7     mitad = lista_tamano // 2
8     resultado[0] = lista[mitad]
9
10    derecha, izquierda = mitad - 1, mitad + 1
11    i = 1
12    while i < lista_tamano:
13        if i % 2 == 1:
14            resultado[i] = lista[derecha]
15            derecha -= 1
16        else:
17            resultado[i] = lista[izquierda]
18            izquierda += 1
19        i += 1
20
21    return resultado
```

2. Ejecutar el algoritmo en un lenguaje de programación y verificar el funcionamiento correcto del mismo.

Main:

```
1 import algoritmo as al
2
3 entrada_usuario = input("Ingresa una lista de números separados por espacios: ")
4 lista_entrada = [int(x) for x in entrada_usuario.split()]
5
6 lista_sorteadada = al.ordenamiento_diferencia_absoluta(lista_entrada)
7 print("Lista ordenada por diferencia absoluta:", lista_sorteadada)
```

Ejecución:

```
PS C:\Users\Dennys\Desktop\Taller11> & C:/Users/Dennys/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Dennys/Desktop/Taller11/leccionComplejidad/main.py
Ingresa una lista de números separados por espacios: 2 5 4 8 6 7 9
Lista ordenada por diferencia absoluta: [6, 5, 7, 4, 8, 2, 9]
PS C:\Users\Dennys\Desktop\Taller11>
```

Conclusión: Este algoritmo ha reorganizado los números para minimizar las diferencias absolutas entre los números adyacentes, lo que puede ser útil en situaciones donde se quiere evitar grandes fluctuaciones entre los valores cercanos.

3. De acuerdo a su criterio:

- ¿Cuál es la diferencia entre programación dinámica y algoritmos voraces?

La principal diferencia es que la programación dinámica involucra la resolución de un problema tratando de dividirlo en problemas más pequeños, en el cual almacena las soluciones anteriores para evitar volver a calcularlos, mientras que los algoritmos voraces se centran en tomar decisiones a cada paso del algoritmo para tomar la solución más optima.

- ¿Cuáles son las ventajas o desventajas de utilizar programación dinámica?

Las principales ventajas es que permite dividir problemas complejos en problemas mas pequeños y más sencillos de resolver, además de que evita hacer cálculos repetidos, en cuanto a las desventajas es que puede requerir mas recursos y más tiempo para su ejecución, además de que no todos los problemas pueden ser resueltos mediante programación dinámica.

- ¿Cuáles son las ventajas o desventajas de utilizar algoritmos voraces?

Las principales ventajas es que son relativamente sencillos de implementar dependiendo del problema, además que pueden proporcionar en la mayoría de casos la solución al problema más factible, en cuanto a las desventajas es que si aumenta la complejidad del problema al algoritmo se le dificulta más llegar a la solución óptima, además que no todos los problemas pueden ser resueltos mediante algoritmos voraces.