

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

MODELOS DISCRETOS PARA LA INGENIERA DEL SOFTWARE

Nombre: Achig Toapanta Steven Jossue

NRC: 8001

ID: L00405925

Fecha: 12/01/2023

1. ParkingBill

Analizamos el problema donde se nos pide calcular el valor total de una factura de estacionamiento con ciertas condiciones, primero definamos nuestras condiciones y variables

Valor a pagar por la entrada: 2

Valor por la primera hora: 3

Valor de 2 horas en adelante (parcial o completas): 4

Sabemos que si se pasan de al menos 1 minuto se cobrara la hora como tal

Entonces sabemos que:

-) Si las restas de los minutos son iguales a cero solo deberemos analizar las horas ya que quiere decir que ha usado el numero de horas exacto lo podemos conseguir restando el numero de horas de salida menos el de horas de entrada
-) Otra consideración es que, si la hora de salida es la misma que la hora de salida automáticamente cobramos solo el precio de entrada y la hora fracción, porque quiere decir que salió en la misma hora sin importar los minutos.
-) Ahora si después de aclarar esos puntos vamos a analizar los minutos
-) Si los minutos de entrada son mayores a los minutos de entrada entonces se cobrará el valor de entrada la primera hora y la resta de las horas de entrada y salida menos 1 y multiplicada por 4
-) Caso contrario simplemente se cobrará el valor de la entrada y la primera hora mas la resta de las horas de salida y entrada multiplicada por 4

2. ParityDegree

Necesitamos encontrar la mayor potencia de dos a la que un numero N puede ser dividido, lo que hacemos es analizar las potencias de 2 con el módulo es decir el residuo entonces mientras el $N \% 2^k == 0$ el k seguirá aumentando hasta que el residuo sea diferente de cero entonces ahí se tomará el k restado 1 como la mayor potencia de dos a la que se puede dividir el número.

Hay que tener claro que esto solo funcionara para numerosa pares ya que los impares no pueden ser divididos de manera entera para 2.

3. FirstUnique

Lo que necesitamos es saber cual es el primer elemento en repetirse en una arreglo de números

Con una función de python podemos analizar la cadena original y los elementos que no se repiten así si el tamaño de la cadena original es igual al de la cadena con los elementos sin repetir eso quiere decir que no hay números repetidos caso contrario deberemos analizar cual índice por índice hasta que uno sea igual y el primer caso que aparezca ese será la respuesta.

Repositorio:

https://github.com/StevenJossueAchig/-NRC_8001-_Repositorio_Achig_Steven