



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ingeniería

Profesor: M.I. Oscar René Valdez Casillas

Fundamentos de Programación

Práctica 03: Solución de problemas y algoritmos

Integrantes:

Arteaga Gonzalez Jaime Alejandro

Márquez Alan

Valencia Francisco Valentín

García Jiménez Joel David

Grupo: 31

Arteaga, J., Márquez, A., García, J. y Valencia, V., (2024). Práctica 03: Solución de problemas y Algoritmos.

Índice:

| | |
|---|----------|
| 1. Resumen | 2 |
| 2. Introducción | 2 |
| • Descripción general | |
| 3. Objetivo y desarrollo de contenidos | 2 |
| 4. Ejercicio 1. Algoritmo para calcular el término 10 de la secuencia de Fibonacci | 2 |
| • Entrada: Es la introducción de datos que el algoritmo necesita para operar. | |
| • Proceso: Es el conjunto de operaciones que se realizan para resolver el problema. | |
| • Salida: Son los resultados que se obtienen después de que el algoritmo termina su ejecución. | |
| 5. Ejercicio 2. Algoritmo de boletos en un estadio | 3 |
| • Entrada: Es la introducción de datos que el algoritmo necesita para operar. | |
| • Proceso: Es el conjunto de operaciones que se realizan para resolver el problema. | |
| • Salida: Son los resultados que se obtienen después de que el algoritmo termina su ejecución. | |
| 6. Ejercicio 3. División de números positivos enteros y primos | 5 |
| • Entrada: Es la introducción de datos que el algoritmo necesita para operar. | |
| • Proceso: Es el conjunto de operaciones que se realizan para resolver el problema. | |
| • Salida: Son los resultados que se obtienen después de que el algoritmo termina su ejecución. | |
| 7. Conclusiones..... | 6 |
| • Reflexión individuales de los integrantes del equipo | |
| 8. Bibliografía..... | 7 |
| • Fuentes y referencias utilizadas en el informe | |

Resumen

Se han realizado problemas para la identificación del conjunto de entradas y salidas, tomando en cuenta habilidades analíticas y de diseño de los algoritmos. A través de diversos problemas se busca el correcto conocimiento de módulos de entrada, procesamiento y salida mediante la creación de algoritmos que brindan soluciones eficientes con los conocimientos del ciclo de vida de un software.

Introducción

La elaboración de algoritmos comienza con un análisis mediante el cual se identifican las necesidades del usuario, los conjuntos de entrada y de salida de tal forma que se pueda diseñar una resolución a la problemática con un algoritmo. Tomando en cuenta que el algoritmo forma parte de la etapa de diseño en el ciclo de vida de un software, a continuación se le darán soluciones a distintos problemas mediante alternativas óptimas de un algoritmo.

Objetivo: Elaborar algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas de Análisis y Diseño pertenecientes al Ciclo de vida del software.

Desarrollo de contenidos:

- A partir del enunciado de un problema, identificar el conjunto de entrada y el conjunto de salida.
- Elaborar un algoritmo que resuelva un problema determinado (dado por el profesor), identificando los módulos de entrada, de procesamiento y de salida.

Ejercicio 1. Encuentre el algoritmo para calcular el término 10 de la secuencia de Fibonacci. Recuerde que los dos primeros números de la serie son 0 y 1. El resto se calcula con la suma de los dos números inmediatos que le preceden.

Conjunto de Entrada:

- No se requiere ninguna entrada del usuario, ya que se pide calcular el término 10 de la secuencia de Fibonacci.

Conjunto de Salida:

- El término 10 de la secuencia de Fibonacci.

Algoritmo para calcular el término 10 de la secuencia de Fibonacci

1. Módulo de Entrada:

- Inicializar los dos primeros términos de la secuencia 0 y 1.

2. Módulo de Procesamiento:

- Usar un bucle para calcular los siguientes términos de la secuencia hasta llegar al término 10.
- Guardar el valor del término actual como la suma de los dos términos anteriores.

3. Módulo de Salida:

- Mostrar o devolver el valor del término 10 de la secuencia.

Ejercicio 2. En un estadio se tienen 5 tipos diferentes de localidades, las cuales se identifican por una clave numérica que es un valor comprendido entre 1 y 5. Los precios de cada localidad y los datos referentes a las ventas de boletos para el próximo juego son:

Clave 1 → Boleto 1 P1 → Precio boleto 1

Clave 2 → Boleto 2 P2 → Precio boleto 2

Clave 3 → Boleto 3 P3 → Precio boleto 3

Clave 4 → Boleto 4 P4 → Precio boleto 4

Clave 5 → Boleto 5 P5 → Precio boleto 5

Algoritmo:

```
// Declaración de variables: Entradas
```

```
ENTERO P1, P2, P3, P4, P5
```

```
ENTERO clave, cantidad, importe
```

```
ENTERO cantidad_vendida1 = 0, cantidad_vendida2 = 0, cantidad_vendida3 = 0,  
cantidad_vendida4 = 0, cantidad_vendida5 = 0
```

```
ENTERO recaudacion_total = 0
```

```
// a) Leer los precios de los boletos
```

```
IMPRIMIR "Ingrese el precio de los boletos para cada clave"
```

```
LEER P1, P2, P3, P4, P5
```

// b) Leer las ventas de boletos

MIENTRAS haya ventas HACER

IMPRIMIR "Ingrese la clave del boleto vendido (1-5) y la cantidad de boletos vendidos"

LEER clave, cantidad

// c) Calcular el importe de la venta según la clave y actualizar la cantidad vendida

SEGÚN clave HACER

CASO 1:

$\text{importe} = \text{cantidad} * P1$

$\text{cantidad_vendida1} = \text{cantidad_vendida1} + \text{cantidad}$

CASO 2:

$\text{importe} = \text{cantidad} * P2$

$\text{cantidad_vendida2} = \text{cantidad_vendida2} + \text{cantidad}$

CASO 3:

$\text{importe} = \text{cantidad} * P3$

$\text{cantidad_vendida3} = \text{cantidad_vendida3} + \text{cantidad}$

CASO 4:

$\text{importe} = \text{cantidad} * P4$

$\text{cantidad_vendida4} = \text{cantidad_vendida4} + \text{cantidad}$

CASO 5:

$\text{importe} = \text{cantidad} * P5$

$\text{cantidad_vendida5} = \text{cantidad_vendida5} + \text{cantidad}$

FIN CASOS

// c) Imprimir clave, cantidad e importe de la venta

IMPRIMIR "Clave: ", clave, " Cantidad: ", cantidad, " Importe: ", importe

// Actualizar la recaudación total

$\text{recaudacion_total} = \text{recaudacion_total} + \text{importe}$

FIN MIENTRAS

// d) Imprimir la cantidad de boletos vendidos de cada tipo

IMPRIMIR "Boletos vendidos - Clave 1: ", cantidad_vendida1

```

IMPRIMIR "Boletos vendidos - Clave 2: ", cantidad_vendida2
IMPRIMIR "Boletos vendidos - Clave 3: ", cantidad_vendida3
IMPRIMIR "Boletos vendidos - Clave 4: ", cantidad_vendida4
IMPRIMIR "Boletos vendidos - Clave 5: ", cantidad_vendida5

```

```
// e) Imprimir la recaudación total del estadio
```

```
IMPRIMIR "Recaudación total: ", recaudacion_total
```

```
FIN ALGORITMO
```

Ejercicio 3. Un número positivo N es un número primo si los únicos enteros positivos que lo dividen son exactamente 1 y N . Indique la forma en que dado un número M , se obtenga y cuente todos los números primos menores a M .

```
ALGORITMO ContarNumerosPrimos
```

```
// Declaración de variables
```

```
ENTERO M, i, j, contador_primos = 0
```

```
BOOLEANO es_primo
```

```
// Leer el valor de M
```

```
IMPRIMIR "Ingrese el valor de M:"
```

```
LEER M
```

```
// Recorrer todos los números desde 2 hasta M-1
```

```
PARA i = 2 HASTA M-1 HACER
```

```
    // Suponemos que i es primo
```

```
    es_primo = VERDADERO
```

```
    // Verificar si i tiene divisores diferentes de 1 y de sí mismo
```

```
    PARA j = 2 HASTA i-1 HACER
```

```
        SI i MOD j == 0 ENTONCES
```

```
            es_primo = FALSO
```

```
            ROMPER // No es necesario seguir verificando
```

```
        FIN SI
```

```
    FIN PARA
```

```

// Si es_primo sigue siendo VERDADERO, entonces i es primo
SI es_primo ENTONCES
    IMPRIMIR i, " es un número primo."
    contador_primos = contador_primos + 1
FIN SI
FIN PARA

// Imprimir el total de números primos encontrados
IMPRIMIR "Total de números primos menores a ", M, " es: ", contador_primos
FIN ALGORITMO

```

Conclusiones:

Jaime: El hacer un algoritmo no es nada sencillo, para muchos posiblemente sea lo más fácil cuando se tiene que programar o diseñar un programa, pero tengamos en cuenta que el hacer esto implica una relevancia para el resto del proceso de la creación del programa, a que me refiero con esto?, me refiero a que antes de pasar todo esto a la computadora, es necesario tener un buen algoritmo, tener una buena guía y por así decirlo una receta de lo que vayas a hacer, tener los pasos para la creación de un programa, que posteriormente lo pasará a una computadora.

Esta práctica nos ayudó a hacer algoritmos en diferentes casos y con lo que nos había enseñado el profesor, se logró obtener algoritmos en el que especificamos todas las variables, de entrada, salida y sobre todo el proceso. Me llevo de esta práctica, el aprendizaje más a fondo de los algoritmos y un buen previo hacia lo que se viene, es decir programar estos algoritmos en la computadora, ahora si ponerlo en práctica, y así sabrás si el algoritmo fue bien hecho y de manera correcta.

Joel: En esta práctica, se abordaron problemas orientados a la identificación y resolución de algoritmos, permitiendo desarrollar habilidades analíticas y de diseño. Se enfatizó en la correcta definición de los módulos de entrada, procesamiento y salida, elementos clave en la creación de algoritmos eficientes y funcionales, lo cual es esencial en las etapas de análisis y diseño del ciclo de vida de un software. Los ejercicios propuestos permitieron poner en práctica estos conceptos, asegurando una comprensión sólida del proceso de desarrollo de algoritmos, desde la identificación de las entradas y salidas hasta la implementación de soluciones óptimas.

Alan: En esta práctica, se han desarrollado algoritmos que permiten resolver problemas específicos de manera estructurada y eficiente. El proceso de diseñar un algoritmo involucra descomponer el problema en pasos lógicos y secuenciales, lo cual facilita su implementación en un lenguaje de programación.

La secuencia de Fibonacci es un ejemplo clásico de cómo un problema aparentemente sencillo puede ser resuelto de manera eficiente mediante la correcta aplicación de un algoritmo. En este caso, se utilizó una estrategia iterativa para calcular cada término de la secuencia a partir de los dos términos anteriores, lo que demostró la importancia de comprender el flujo de control y las estructuras de repetición en programación.

Estos ejercicios refuerzan la habilidad de crear algoritmos eficientes, que son fundamentales en la programación para resolver problemas complejos de manera efectiva. Además, este tipo de actividades fomenta el pensamiento lógico y la capacidad de transformar ideas abstractas en soluciones concretas.

Valentín: El desarrollo de algoritmos es un proceso crucial y complejo en la creación de software, ya que establece la base sobre la cual se construirá el programa final. Aunque algunos consideran que diseñar un algoritmo es la parte más sencilla de la programación, es esencial entender que un buen algoritmo es fundamental para guiar el resto del proceso de desarrollo. Antes de traducir las ideas a código, es necesario contar con una estructura clara y bien definida, que actúe como una receta detallada para la implementación.

Durante la práctica, se trabajó en la identificación y resolución de problemas a través del diseño de algoritmos, lo cual permitió desarrollar habilidades clave. Se puso especial énfasis en la correcta definición de los módulos de entrada, procesamiento y salida, que son esenciales para la creación de algoritmos eficientes. Estas actividades ayudaron a comprender mejor el proceso de desarrollo de algoritmos. Como ejemplo se abordó la secuencia de Fibonacci, que demostró cómo un problema aparentemente sencillo puede resolverse de manera eficiente mediante una estrategia iterativa.

Referencias:

Algoritmos: qué son y qué tipos existen - Ferrovial. (2022, 18 octubre). Ferrovial.

<https://www.ferrovial.com/es/stem/algoritmos/#:~:text=Algoritmos%20de%20b%C3%BAqueda:%20aquellos%20que,para%20determinar%20si%20son%20iguales.>

Equipo editorial, Etecé. (2023, 15 febrero). Ejemplos de algoritmos - Qué son, tipos y características. Concepto.

<https://concepto.de/ejemplos-de-algoritmos/#:~:text=Este%20algoritmo%20tan%20cotidiano%20maneja,m%C3%BAltiples%20sistemas%20de%20datos%20simult%C3%A1neos.>