



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Adrian Ulises Mercado Martinez

Profesor:

Fundamentos de programación

Asignatura:

6

Grupo:

Práctica 3

No de Práctica(s):

Diego Emmanuel Ibarra Laureano

Integrante(s):

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada:

2021-1

Semestre:

29/10/20

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

Índice:

Objetivo de la práctica:.....pág. 1

Introducción:.....pág. 1 - pág.2

Desarrollo:.....pág.3 - pág.5

Conclusiones:.....pág.5

Bibliografías:.....pág.5

Solución de problemas y algoritmos

Objetivo de la práctica:

Elaborar algoritmos correctos y eficientes en la solución de problemas siguiendo los pasos de Análisis y Diseño pertenecientes al ciclo de vida del software.

Introducción:

En las ciencias de la computación se ocupa de los problemas computables. Se le llama problema computable a aquella abstracción de la realidad que tiene representación algorítmica. Los algoritmos permiten encontrar la solución a problemas computables.

Un algoritmo se puede definir como una secuencia de instrucciones que representan un modelo de solución para determinado tipo de problemas. O bien un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema.

Características de los algoritmos:

- . Preciso: Definirse de manera rigurosa, sin dar lugar a ambigüedades.
- . Definido: Si un algoritmo es repetido dos veces tendrá el mismo resultado.
- . Finito: Debe terminar en algún momento.
- . Puede tener 0 o más elementos de entrada.
- . Debe producir un resultado. Los datos de salida serán resultados de efectuar el algoritmo.

La Ingeniería de Software provee métodos que indican cómo generar software. Estos métodos abarcan una amplia gama de tareas:

Planeación y estimación del proyecto.

Análisis de requerimientos del sistema y software.

Diseño de la estructura de datos, la arquitectura del programa y el procedimiento algorítmico.

Codificación

Ciclo de vida del software:

El término ciclo de vida del software describe el desarrollo de software, desde la fase inicial hasta la fase final. El propósito de este programa es definir las distintas fases intermedias que se requieren para validar el desarrollo de la aplicación, es decir, para garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo: se asegura de que los métodos utilizados son apropiados.

En otras palabras, el proceso que se sigue para construir, entregar y hacer evolucionar el software, desde la concepción de una idea hasta la entrega y retiro del sistema. Se definen las distintas fases intermedias que se requieren para validar el desarrollo de un software, es decir, para garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo, se asegura de que los métodos utilizados son apropiados.



Figura 1: Ciclo de vida del software.

Solución de problemas:

La resolución de problemas utilizando como herramienta una computadora no se resume únicamente en la escritura de un programa, sino que se trata de una tarea más compleja. El proceso abarca todos los aspectos que van desde interpretar las necesidades del usuario hasta verificar que la respuesta brindada es correcta. Las etapas son las siguientes:

Análisis del problema

En esta primera etapa, se analiza el problema en su contexto del mundo real. Deben obtenerse los requerimientos del usuario. El resultado de este análisis es un modelo preciso del ambiente del problema y del objetivo a resolver. Dos componentes importantes de este modelo son los datos a utilizar y las transformaciones de los mismos que llevan al objetivo.

Diseño de una solución

La resolución de un problema suele ser una tarea muy compleja para ser analizada como un todo. Una técnica de diseño en la resolución de problemas consiste en la identificación de las partes (subproblemas) que componen el problema y la manera en que se relacionan. Cada uno de estos subproblemas debe tener un objetivo específico, es decir, debe resolver una parte del problema original. La integración de las soluciones de los subproblemas es lo que permitirá obtener la solución buscada.

Especificación de algoritmos

La solución de cada subproblema debe ser especificada a través de un algoritmo. Esta etapa busca obtener la secuencia de pasos a seguir para resolver el problema. La elección del algoritmo adecuado es fundamental para garantizar la eficiencia de la solución.

Escritura de programas

Un algoritmo es una especificación simbólica que debe convertirse en un programa real sobre un lenguaje de programación concreto. A su vez, un programa escrito en un lenguaje de programación determinado (ej: Pascal, Ada, etc) es traducido automáticamente al lenguaje de máquina de la computadora que lo va a ejecutar. Esta traducción, denominada compilación, permite detectar y corregir los errores sintácticos que se cometan en la escritura del programa.

Verificación

Una vez que se tiene un programa escrito en un lenguaje de programación se debe verificar que su ejecución produce el resultado deseado, utilizando datos representativos del problema real. Sería deseable poder afirmar que el programa cumple con los objetivos para los cuales fue creado, más allá de los datos particulares de una ejecución. Sin embargo, en los casos reales es muy difícil realizar una verificación exhaustiva de todas las posibles condiciones de ejecución de un sistema de software. La facilidad de verificación y la depuración de errores de funcionamiento del programa conducen a una mejor calidad del sistema y es un objetivo central de la Ingeniería de Software.

Desarrollo de la práctica:

Problema 1: Algoritmo para la solución del producto de dos matrices.

F8										
	A	B	C	D	E	F	G	H	I	J
1		Columna1	Columna2	Columna3		Columna1	Columna2	Columna3		
2	Renglón 1	1	3	1	Renglón 1	1	3	1		
3	Renglón 2	2	2	2	Renglón 2	2	2	2		
4	Renglón3	3	1	3	Renglón3	3	1	3		
5		A	3x3			B	3x3			
6				Columna1	Columna2	Columna3				
7			Renglón 1	10	10	10		C=AxB		BxA
8			Renglón 2	12	12	12				
9			Renglón3	14	14	14				
10										
11	3	x	3		3	x	3			
12			3	x	3					
13										
14										
15	2	x	4		4	x	5			
16										
17			2	x	5					
18										
19										

Imagen 2: Como hacer una multiplicación de dos matrices.

Dos matrices se pueden multiplicar. Solo si el número de columnas de la primera matriz es igual al número de renglones de la segunda(ejemplo en la imagen 2).

El producto de la multiplicación de dos matrices es igual a la multiplicación y suma de los valores de cada columna y renglon de las dos matrices.

P1: Proponer un algoritmo para la solución de una multiplicación de dos matrices de n número de columnas y renglones. Indicando los modulos de entrada, procesamiento y salida.

Entrada Dos matrices de n # de columnas y n # de renglones.

Salida: Una matriz resultante.

Restricciones: Si el número de columnas de la matriz a es diferente al número de renglones de la matriz b entonces no se hace la multiplicación.

1. Si el número de columnas de A es igual al número de renglones de B
 - 1.1 Iniciar i en cero
 - 1.2 iniciar j en uno
 - 1.3 Iniciar K en uno
 - 1.4 Hacer suma igual a cero
 - 1.5 Hacer a igual a la renglón i
 - 1.6 Hacer b igual a la columna j
 - 1.7 Hacer suma igual suma más el producto= suma + a[K] * b[K]
2. Hacer C igual a la matriz resultante
 - 2.1 Hacer x igual a los renglones de C
 - 2.2 Hacer y igual a las columnas de C
 - 2.3 Iniciar x en cero
 - 2.4 Iniciar y en cero
 - 2.5 Hacer suma igual a cero
 - 2.6 Colocar los resultado de Suma en K en x(n) y y(n)
 - 2.7 Imprimir Matriz resultante C
 - 2.8 Fin

1 Si el número de columnas de A es igual al número renglones de B	
1.1 Iniciar i en uno	
1.2 Iniciar j en uno	
1.3 Iniciar k en uno	
1.4 Hacer suma igual cero	
1.5 Hacer a igual al renglón i	
1.6 Hacer b igual a la columna j	
1.7 Hacer suma igual suma más el producto $a[k]*b[k]$	$\text{suma}=\text{suma}+a[k]*b[k]$

Imagen 3: Algoritmo de operación Matriz x Matriz.

Pruebas de escritorio:

1

4	1	8
7	3	1
9	3	5

2	4	1
3	5	9

Matriz resultante:

El número de columnas de la primera matriz es diferente al número de renglones de la segunda, por lo tanto no puede hacerse la multiplicación.

2

2	6
1	7
9	1

3	4	6
2	6	3

Matriz resultante:

18	18	30
17	46	27
29	42	57

10	9	2
1	12	1
4	15	3
7	7	4

1	4	3	8
9	1	17	5
4	16	2	7

Matriz resultante:

El número de columnas de la primera matriz es diferente al número de renglones de la segunda, por lo tanto no puede hacerse la multiplicación.

Conluciones:

En la práctica observamos como se construye un algoritmo con respecto a una serie de pasos a seguir originados por un problema matemático en este caso, más específicamente la resolución de una multiplicación de dos matrices.

Los algoritmos nos pueden ayudar a plantear o proponer soluciones de algún problema de manera rigurosa, con un número de acciones finitas y siempre con el orden que establece el problema. Y esto es gracias a el análisis y todos los pasos que nos proporciona el Ciclo de vida del Software.

Bibliografías:

. Ángel Robledano. (2019). Que es un algoritmo informático. 25/10/20, de OpenWebinars Sitio web: <https://openwebinars.net/blog/que-es-un-algoritmo-informatico/>

. Ivan Ramirez . (2018). Ciclo de vida de desarrollo de software. 25/10/20, de Efecto Digital Sitio web: <https://www.efectodigital.online/post/2018/04/23/ciclo-de-vida-de-desarrollo-de-software>

. Mtro. Luis Eduardo Pérez Bernal. (2008). Algoritmos y solucion de problemas . 25/10/20, de ITESO Sitio web: <http://recursos.salonesvirtuales.com/wp-content/uploads/bloques/2012/07/AlgoritmoSoluciondeProblema.pdf>