
¿Qué es un algoritmo?

M. en C. Paola Nayeli Cortez Herrera

M en C. Jorge Fonseca Campos

M. en C. Carlos de la Cruz Sosa

Instituto Politécnico Nacional

Unidad Profesional en Ingeniería y Tecnologías Avanzadas (IPN- UPIITA)

Resumen

El concepto de algoritmo va ligado a la vida diaria. Pues de acuerdo con la definición del concepto, diario se ejecutan una serie de pasos finitos para realizar las actividades cotidianas de principio a fin. En el ámbito de la computación, en particular cuando se trata del área de programación, el concepto de algoritmo toma un significado más complejo.

Muchas veces en los cursos introductorios de programación el tema de algoritmo es visto de forma breve, pues en la mayoría de las ocasiones el profesor se centra más en enseñar la sintaxis del lenguaje de programación, que profundizar en el análisis para que el alumno sea capaz de comprender a detalle el concepto de algoritmo y sobre todo diseñar uno. Esto conlleva a que en la actualidad al momento de revisar propuestas de tesis de licenciatura se encuentre como parte del título o los objetivos el proponer o desarrollar algoritmos, sin embargo no se le da la importancia a esta acción.

Razón por la cual este trabajo se encarga de proporcionar algunas definiciones e ideas relacionadas al proponer y desarrollar algoritmos.

Introducción

A diario se desarrollan un conjunto de pasos para realizar ciertas acciones. Pero de forma consiente no se lleva un registro de estos pasos. Al conjunto de pasos para lograr ciertas acciones se denomina algoritmo. Entre algunas definiciones de algoritmo se encuentran:

- Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema [6].
- Conjunto de reglas para efectuar algún cálculo, bien sea a mano o, más frecuentemente, en una máquina [1].

Al proponer el desarrollo de un algoritmo en muchas ocasiones se olvida el proceso que esta acción implica, y la persona se enfoca principalmente en la forma en que codificará pasando por alto el análisis minucioso del problema a resolver. Esto es, identifica la salida del problema pero como ya se encuentra programando hace uso de 'n' entradas, que si hubiera definido desde un principio estarían bien identificadas.

Y es que como se menciona en [4], la escritura de programas es la elaboración o implementación de algoritmos. Al final, programar bien es hacer matemáticas y, como todo buen matemático, todo buen programador debe contar con un buen método (formal o intuitivo) que le permita asegurarse de que sus programas son correctos. Aquí es donde empieza a verse la importancia de la definición y concepción del término algoritmo.

Importancia de un algoritmo

Algo que en muchas ocasiones se pasa por alto al momento de empezar a proponer un algoritmo es que como se menciona en [2] los algoritmos son independientes tanto del lenguaje de programación en que se expresan como del equipo de cómputo que los ejecuta.

Esto provoca que al tener ya elegido un lenguaje de programación previo a la propuesta del algoritmo, este se vea afectado en su diseño. Situación que ocurre la mayoría de las veces cuando los alumnos proponen sus temas de tesis y ya tienen determinado el lenguaje de programación utilizarán.

En [5], se mencionan los pasos para la solución de un problema a través de la computadora son: definición del problema, análisis del problema, crear el algoritmo, y la prueba de escritorio. En donde se nota que para llegar a la creación del algoritmo es necesario tener bien definido el problema y analizarlo.

Pasos en la construcción del algoritmo

Una vez que ya se tiene el problema bien definido es momento de empezar a trabajar en el algoritmo que le dé la solución. A continuación se proponen los siguientes de pasos tomando como base la experiencia al asesorar trabajos de tesis de licenciatura.

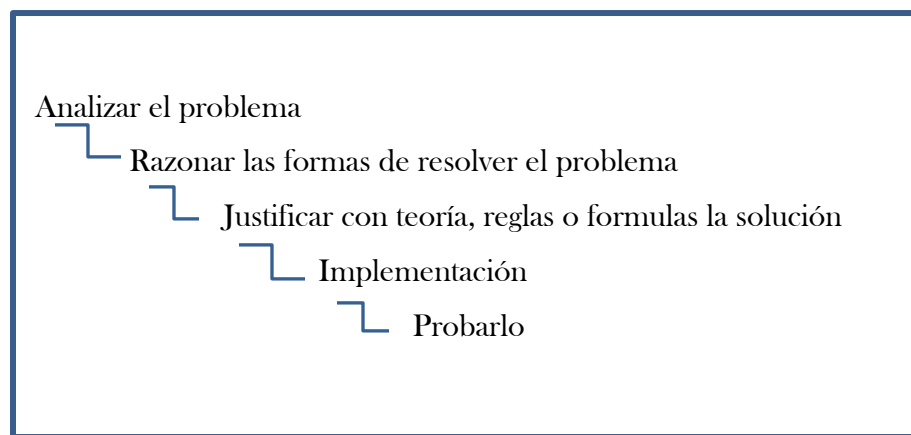


Figura 1. Fases a considerar en el diseño de un algoritmo.

En la figura anterior, se resaltan los conjuntos de pasos que deben ser considerados al momento de hacer la propuesta de un algoritmo, a continuación se detallan:

1. Analizar el problema. Este es el punto de partida al momento de pretender dar solución a una problemática. Básicamente consiste en responder: ¿Qué se va a resolver?, ¿Qué resultados se esperan?, ¿Para quién van dirigidos los resultados?, ¿Qué se necesita para llegar a esos resultados? Cuando este paso se cumple se logran identificar todas las variables y factores que intervendrán en el algoritmo, inclusive identificar los diversos escenarios que se contemplarán.

2. Razonar las formas de resolver el problema. Cuando ya se tienen los elementos básicos del problema o situación a resolver, es necesario identificar cuantas posibles soluciones tiene. Aquí la lógica de cada persona influye al momento de encontrar la solución. Para este paso es necesario estar documentado sobre cómo han sido resueltos problemas similares, para que sirva de guía.
3. Justificar con teoría, reglas o formulas la solución. De la 'n' cantidad de soluciones posibles, ¿Cómo saber cuál es la más acertada? En definitiva para responder esta pregunta se necesitará aplicar rigor a la solución seleccionada. Quizás mientras una solución implica un promedio de diversos valores, la otra solución utiliza la probabilidad para encontrar un rango más acertado de valores. O mientras una solución implica únicamente 3 variables de entrada, la otra usa 7 variables y permite tener mayor nivel de detalle. Áreas como las matemáticas, estadísticas, análisis numérico, entre otras deben de ser usadas a fin de justificar la solución.
4. Implementación. En este punto es donde se empieza a pensar cuál lenguaje de programación será idóneo para codificar el algoritmo. El proceso de codificación se realiza en este paso. Si el algoritmo involucra varias iteraciones de cálculos matemáticos, seguramente un lenguaje como C, C++, o Python podrán ser algunos lenguajes de programación que se deberán de elegir. O si el algoritmo involucra gran uso de expresiones regulares entonces lenguajes de programación como Perl, quizás C o Java podrían ser opción.
5. Probarlo. El momento decisivo en la propuesta del algoritmo, verificar que la solución que encontramos como idónea realmente arroje los resultados esperados. En este paso es cuando se verifica si se hizo un correcto análisis del problema. Es posible realizar un leve ajuste a algunos parámetros del algoritmo, con la finalidad de mejorar las salidas.

Un error encontrado al proponer un algoritmo es que realmente el programador no comprende ¿Cuál es la problemática a resolver? En muchas ocasiones el programador todo lo ve como código pero sin profundizar en el análisis del problema.

Características del algoritmo

Un algoritmo debe de cumplir ciertas características, de acuerdo con [3] entre las características fundamentales están:

- Un algoritmo debe ser **preciso** e indicar el orden de realización de cada paso.
- Un algoritmo debe estar **definido**. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- Un algoritmo debe ser **finito**. Si se sigue un algoritmo, se debe terminar en algún momento; o sea, debe tener un número finito de pasos.

En [4], se agregan otras características las cuales son:

Tener cero o más entradas. Identificar qué datos son proporcionados a un algoritmo como insumo (o estos son generados de alguna forma), para llevar a cabo las operaciones que comprende.

Tener una o más salidas. Debe siempre devolver un resultado; de nada sirve un algoritmo que hace algo y nunca sabemos que fue. El devolver un resultado no debe ser considerado como únicamente “verlos” en forma impresa o en pantalla, como ocurre con las computadoras.

Efectividad. El tiempo y esfuerzo por cada paso realizado debe ser preciso, no usando nada más ni nada menos que aquello que se requiera para y en su ejecución.

Para reflexionar

La principal motivación de presentar un breve panorama sobre los algoritmos fue el comentario que escuché de parte de unos alumnos: *“el algoritmo es demasiado sencillo, solo son if’s y case’s”*.

En esos momentos comprendí que como asesora había dado por hecho que ellos en el noveno semestre de su carrera y habiendo cursado materias de programación sabían qué era un algoritmo. El comentario también hace que se reflexione sobre la forma en que se da la clase, quizás como profesora de programación me he enfocado más en enseñarles sintaxis, que trabajar en la lógica para resolver problemas.

Conclusiones

La solución de problemas basados en un algoritmo detallado y con un análisis previo, seguramente dará resultados satisfactorios. Sin embargo el problema al que muchas propuestas de tesis de licenciatura se enfrentan es que no se sabe a detalle en primero: ¿qué es un algoritmo?, ¿en qué consiste?, y en segundo ¿Qué se quiere resolver? Por lo tanto los algoritmos que se proponen en los trabajos de tesis en muchas ocasiones se encuentran carentes de bases matemáticas, estadísticas e inclusive lógica que lo fundamenten.

Algo que se ha asumido al trabajar con alumnos tesisistas es que ellos saben analizar a detalle un problema para encontrarle una solución, cosa que es falsa pues todavía siguen asumiendo que el asesor les dará las instrucciones para que ellos únicamente programen. O peor aún, el alumno se encuentra confundido al no saber identificar el proceso de codificación con algoritmo.

Razón por la cual este breve escrito refleja parte de los conceptos y etapas básicas para la construcción y desarrollo de un algoritmo.

Bibliografía

- [1]Brassard, G. / Bratley, P.. (2000). Fundamentos de Algoritmia. Madrid, España: Prentice Hall.
- [2]José E. Gallardo Ruiz, Carmen M. García Lopez. (ND). Tema 2. Diseño de algoritmos y programas. En Apuntes para la asignatura Informática (2-13). Universidad De Málaga: Facultad de Ciencias (Matemáticas).
- [3]Juan Bernardo Vázquez Gómez Red Tercer. (2012). Análisis Y Diseño De Algoritmos. CDMX: Red Tercer Milenio.
- [4]M. en C. Eduardo René Rodríguez Ávila. (2016). El Correcto y Completo Desarrollo de un Algoritmo. 04 Abril 2017, de Sección de Estudios de Posgrado e Investigación UPIICSA Sitio web: <https://pastranamoreno.files.wordpress.com/2010/09/algoritmos-hanoi.pdf>
- [5] Oviedo Regino, Efraín M. (2004). Lógica de programación. Bogotá: Colección: Textos Universitarios.
- [6]Real Academia Española. (2017). Definición. 17 Abril 2017, de Real Academia Española Sitio web: <http://dle.rae.es/srv/search?m=30&w=algoritmo>