

Algoritmos y estructuras de Datos

P2PU

Presentación y motivación

Semana 1

Dictado por Marco González Núñez
31 de Enero de 2011

Algoritmos y estructuras de Datos

P2PU

¿Qué es un algoritmo?

Vamos a decir que un algoritmo consta de un conjunto finito de reglas formales ejecutadas de forma secuencial, el cual nos permite obtener una solución a uno o más problemas dados.

Lo que yo llamo: “**Una caja negra**”, en la cual ingresamos un valor (input) y obtenemos un resultado (output).

INPUT

Algoritmo

OUTPUT

Algoritmos y estructuras de Datos

P2PU

Clasificación

Algoritmo determinista: en cada paso del algoritmo se determina de forma única el siguiente paso.

Algoritmo no determinista: deben decidir en cada paso de la ejecución entre varias alternativas y agotarlas todas antes de encontrar la solución.

Todo algoritmo tiene una serie de características, entre otras que requiere una serie de recursos, algo que es fundamental considerar a la hora de implementarlos en una máquina. Estos recursos son principalmente:

El tiempo: período transcurrido entre el inicio y la finalización del algoritmo.

La memoria: la cantidad (la medida varía según la máquina) que necesita el algoritmo para su ejecución.

Obviamente, la capacidad y el diseño de la máquina pueden afectar al diseño del algoritmo.

Algoritmos y estructuras de Datos

P2PU

Además debe cumplir con las siguientes condiciones:

Finitud: Significa que debe ser finito, es decir evitar un loop infinito.

Definibilidad: Es importante que no sea ambiguo, es decir que sea entendible cada paso que se esta haciendo, los algoritmos son definidos en un lenguaje formal.

Entrada: Es necesario que existan 0 o más entradas, estos pueden ser: enteros, arreglos, fracciones, palabras, caracteres, etc.

Salida: Las salidas son lo que obtenemos finalmente de nuestro algoritmo, y esta relacionado directamente con las entradas.

Efectividad: Se entiende por esto que una persona sea capaz de realizar el algoritmo de modo exacto y sin ayuda de una máquina en un lapso de tiempo finito.

Algoritmos y estructuras de Datos

P2PU

Los algoritmos por lo general tienen un número de entradas la cual denominaremos N , entonces el algoritmo deberá ser una función en relación a N . En otras palabras podemos decir que la entrada es usada para obtener la salida suponiendo $N > 0$. De esta forma se puede considerar un tiempo de ejecución del algoritmo, que puede ser proporcional a una de estas funciones:

Tiempo de ejecución constante (1): Esto quiere decir que la mayoría de la instrucciones se ejecutan una vez o muy pocas.

Tiempo de ejecución logarítmico ($\log N$): Esto quiere decir que mientras más grande sea N , más tiempo se demorará sin embargo esta demora no se nota demasiado, ya que $\log N$ solo se duplicará cuando N sea N^2 .

Tiempo de ejecución lineal (N): Esto significa que la función entre la cantidad de datos y el tiempo es lineal, es decir que aumentan de forma proporcional.

Algoritmos y estructuras de Datos

P2PU

Tiempo de ejecución cuadrática (N^2): Esto quiere decir que el aumento de las variables hace que aumente al cuadrado el tiempo de ejecución, acá ya estamos hablando de un tiempo demasiado alto.

Tiempo de ejecución cúbica (N^3): Esto quiere decir que el aumento de las variables hace que aumente al cubo el tiempo de ejecución, acá ya estamos hablando de un tiempo demasiado alto.

Para definir la rapidez de un determinado algoritmo usaremos la notación $O(\text{tiempo de ejecución})$, ejemplo: $O(\log N)$.

Esto significa que el peor de los casos es lo que se demorará nuestro algoritmo en ejecutarse, dependiendo de la cantidad y los tipos de datos ingresados.

Algoritmos y estructuras de Datos

P2PU

La importancia de un buen algoritmo

Muchos de nosotros hemos programado en distintos lenguajes, sin embargo generalmente programamos **a prueba y error** (creo que a varios les debe sonar familiar), cambiamos variables, líneas de código de lugar, hasta que nos da como resultado lo que deseamos. Sin embargo muchas veces esos códigos cuando son ejecutados con algunos datos en especial, tendemos a tener problemas, se nos cae el sistema, o no nos da el resultado deseado.

Es por eso que al momento de codificar es muy importante que sepamos lo que estamos haciendo, **debemos dominar el problema**, conocer todos los distintos tipos de datos que podremos enfrentar, además de ser lo suficientemente ordenado para que otros puedan entender su código, es muy importante que programemos pensando **que lo hacemos para otros**, al final esto nos servirá hasta nosotros mismos cuando digamos: "¿Qué era lo que hice aquí?".

Algoritmos y estructuras de Datos

P2PU

Además de saber cuanto es lo que se demora un algoritmo en entregar su solución, también debemos saber **si es posible hacerlo**, matemáticamente podemos dividirlo en dos partes:

Problemas indecidibles: aquellos que no se pueden resolver mediante un algoritmo.

Problemas decidibles: aquellos que cuentan al menos con un algoritmo para su cómputo.

Sin embargo que sea un problema decidable no podemos asegurar que nos entregue una solución, debido a la gran cantidad de operaciones podría demorarse un tiempo no razonable, según esto un problema decidable se separa por:

Intratables: aquellos para los que no es factible obtener su solución.

Tratables: aquellos para los que existe al menos un algoritmo capaz de resolverlo en un tiempo razonable.

Algoritmos y estructuras de Datos

P2PU

Ahora que entendemos que es un algoritmo, y la importancia de la rapidez de ejecución, nombrare algunos algoritmos de interés:

- El algoritmo del mapa de 4 colores
- Las torres de Hanoi
- El problema del vendedor viajero
- Algoritmo de solución del Sudoku
- Problema de las 8 reinas
- Algoritmo para resolver cubo Rubik

Links de Interés

http://es.wikipedia.org/wiki/Teoría_de_grafos

<http://sigt.net/archivo/las-200-variables-del-algoritmo-de-google.shtml>

Algoritmos y estructuras de Datos

P2PU

Tarea de investigación:

Investigue y escriba **el enunciado** de los problemas de la diapositiva anterior (solo redactar el problema que se necesita resolver) de cada uno de ellos, y elija dos de estos y **redacte con sus palabras** un algoritmo que lo resuelva.

Por ejemplo: En este momento tengo hambre.

Problema a resolver: Que se me quite el hambre

Solución del problema: Me pongo zapatos, bajo las escaleras, esquivo a mi perro, camino hacia la cocina, abro el refrigerador, saco jamón, voy a la bolsa del pan, saco un cuchillo, abro el pan, le pongo el jamón, cierro el pan, y me lo como, y se me quita el hambre.

Nota: No es necesario explicar todos los pasos, solo la forma de encontrar la solución, por ejemplo diciendo: "y así sucesivamente hasta resolver el problema".

Algoritmos y estructuras de Datos

P2PU

La tarea debe ser enviada a mi correo electrónico, a más tardar a las 23:59 del día Viernes 4 de Febrero de 2011, hora de Chile, en formato pdf.

Debe tener una portada con el nombre y nick de cada uno(P2PU), letra Arial 16, para títulos y 12 para texto, interlineado 1.5 y no exceder más allá de 3 páginas incluida la portada.

Tarea opcional:

Investigar sobre diagramas de flujo, y crear un algoritmo que represente un día normal en sus vidas, debe incluir:

- Cuando despiertan y se duermen (inicio y fin del algoritmo)
- Comidas
- Un horario, días de la semana.
- Alguna parte a donde van regularmente.

Debe ser lo más detallado posible e Incluir condiciones y por lo menos un loop, el formato de entrega es el mismo al anterior.

Ej: Despierto a las 8, me ducho, me lavo los dientes, si es lunes voy a la Universidad, Si son las 14 almuerzo, mientras no sean las 17 estoy en el trabajo, etc (con diagramas de flujo).

Algoritmos y estructuras de Datos

P2PU

Prguntas para debatir en los foros:

- 1.- ¿Pueden haber dos o más salidas distintas considerando la misma entrada en un algoritmo?
- 2.- ¿Porqué es importante pensar en resolver el algoritmo antes de comenzar a programar?

Frase de la semana: Loco es definido como: "Aquel individuo que hace siempre lo mismo y espera distintos resultados"