

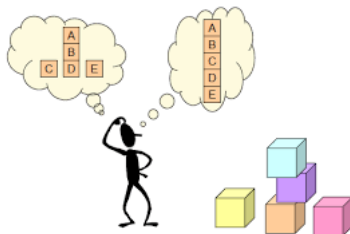
Introducción

Análisis y Diseño de Algoritmos

Dr. Jaime Osorio Ubaldo

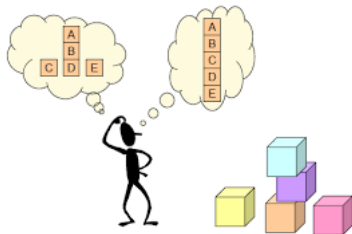
Introducción

- 1 Sabemos que la capacidad de cálculo de los computadores es creciente desde su aparición a finales de los 50.
- 2 Podemos asumir erroneamente, que basta esperar algunos años para que problemas que necesitan muchas horas de cálculo hoy, puedan resolverse en pocos segundos dentro de algunos años.
- 3 Pero en realidad, el tiempo de ejecución depende del algoritmo que se utilice.



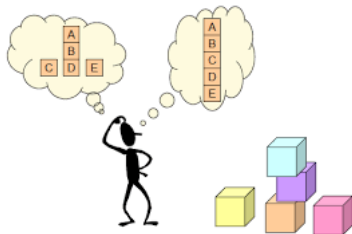
Introducción

- 1 Sabemos que la capacidad de cálculo de los computadores es creciente desde su aparición a finales de los 50.
- 2 Podemos asumir erroneamente, que basta esperar algunos años para que problemas que necesitan muchas horas de cálculo hoy, puedan resolverse en pocos segundos dentro de algunos años.
- 3 Pero en realidad, el tiempo de ejecución depende del algoritmo que se utilice.



Introducción

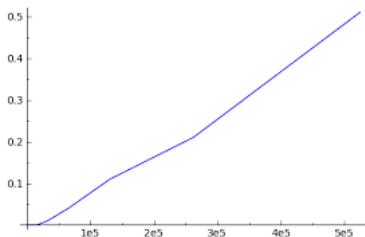
- 1 Sabemos que la capacidad de cálculo de los computadores es creciente desde su aparición a finales de los 50.
- 2 Podemos asumir erroneamente, que basta esperar algunos años para que problemas que necesitan muchas horas de cálculo hoy, puedan resolverse en pocos segundos dentro de algunos años.
- 3 Pero en realidad, el tiempo de ejecución depende del algoritmo que se utilice.



Introducción

Los factores que generalmente influyen en el tiempo de ejecución de un algoritmo.

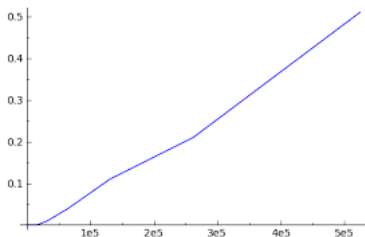
- 1 El contenido de los datos de entrada.
- 2 El código generado por el compilador y computador que se utilice.
- 3 El tamaño de los datos de entrada.



Introducción

Los factores que generalmente influyen en el tiempo de ejecución de un algoritmo.

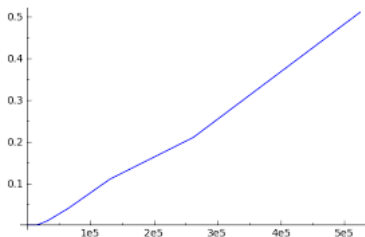
- 1 El contenido de los datos de entrada.
- 2 El código generado por el compilador y computador que se utilice.
- 3 El tamaño de los datos de entrada.



Introducción

Los factores que generalmente influyen en el tiempo de ejecución de un algoritmo.

- 1 El contenido de los datos de entrada.
- 2 El código generado por el compilador y computador que se utilice.
- 3 El tamaño de los datos de entrada.



El contenido de la data.

Comunmente se analiza el algoritmo en el peor caso, es decir, cuando se emplea el mayor tiempo, de tal manera que se tenga una cota superior del tiempo de ejecución para cualquier caso.



El código generado por el compilador y computador que se utilice

Para el análisis de los algoritmos podemos ignorar este factor, ya que diferentes implementaciones de un mismo algoritmo diferirán en sus tiempos de ejecución a lo más en una constante multiplicativa.



Introducción

El tamaño de los datos de entrada. Es el factor más importante .

Ejemplo

En un problema de ordenamiento de n números. Podríamos encontrar computadores cien o quizás mil veces más rápido, pero si el algoritmo empleado es el burbuja el tiempo de ejecución será proporcional a n^2 mientras que de un algoritmo quicksort es proporcional a $n \log n$.

Por lo tanto, el programador debe confiar en sus conocimientos y habilidad para encontrar los mejores algoritmos y no en la capacidad de cálculo de los computadores.



Introducción

El tamaño de los datos de entrada. Es el factor más importante .

Ejemplo

En un problema de ordenamiento de n números. Podríamos encontrar computadores cien o quizás mil veces más rápido, pero si el algoritmo empleado es el burbuja el tiempo de ejecución será proporcional a n^2 mientras que de un algoritmo quicksort es proporcional a $n \log n$.

Por lo tanto, el programador debe confiar en sus conocimientos y habilidad para encontrar los mejores algoritmos y no en la capacidad de cálculo de los computadores.



El tamaño de los datos de entrada. Es el factor más importante .

Ejemplo

En un problema de ordenamiento de n números. Podríamos encontrar computadores cien o quizás mil veces más rápido, pero si el algoritmo empleado es el burbuja el tiempo de ejecución será proporcional a n^2 mientras que de un algoritmo quicksort es proporcional a $n \log n$.

Por lo tanto, el programador debe confiar en sus conocimientos y habilidad para encontrar los mejores algoritmos y no en la capacidad de cálculo de los computadores.



El tamaño de los datos de entrada. Es el factor más importante .

Ejemplo

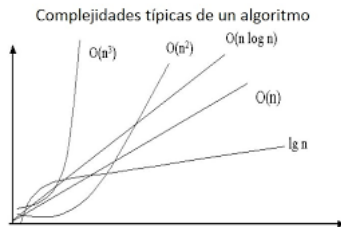
En un problema de ordenamiento de n números. Podríamos encontrar computadores cien o quizás mil veces más rápido, pero si el algoritmo empleado es el burbuja el tiempo de ejecución será proporcional a n^2 mientras que de un algoritmo quicksort es proporcional a $n \log n$.

Por lo tanto, el programador debe confiar en sus conocimientos y habilidad para encontrar los mejores algoritmos y no en la capacidad de cálculo de los computadores.



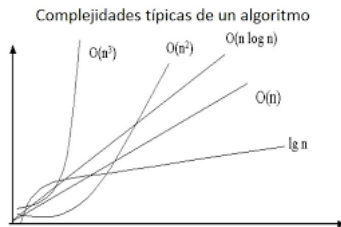
Tipos de complejidad de un algoritmo

- 1 Resolver una ecuación cuadrática $O(1)$
- 2 Búsqueda lineal $O(n)$.
- 3 Recorrido de una matriz $O(n^2)$
- 4 Búsqueda binaria $O(\log(n))$
- 5 Agente viajero : visitar n países en el menor tiempo posible $O(n!)$.



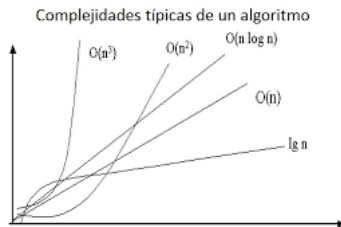
Tipos de complejidad de un algoritmo

- 1 Resolver una ecuación cuadrática $O(1)$
- 2 Búsqueda lineal $O(n)$.
- 3 Recorrido de una matriz $O(n^2)$
- 4 Búsqueda binaria $O(\log(n))$
- 5 Agente viajero : visitar n países en el menor tiempo posible $O(n!)$.



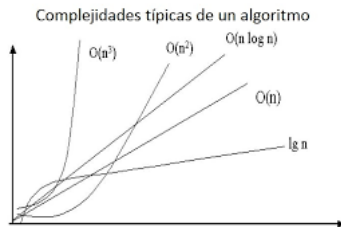
Tipos de complejidad de un algoritmo

- 1 Resolver una ecuación cuadrática $O(1)$
- 2 Búsqueda lineal $O(n)$.
- 3 Recorrido de una matriz $O(n^2)$
- 4 Búsqueda binaria $O(\log(n))$
- 5 Agente viajero : visitar n países en el menor tiempo posible $O(n!)$.



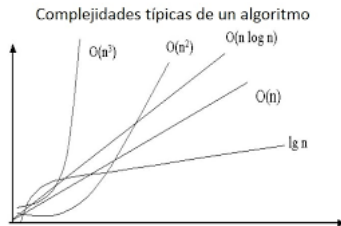
Tipos de complejidad de un algoritmo

- 1 Resolver una ecuación cuadrática $O(1)$
- 2 Búsqueda lineal $O(n)$.
- 3 Recorrido de una matriz $O(n^2)$
- 4 Búsqueda binaria $O(\log(n))$
- 5 Agente viajero : visitar n países en el menor tiempo posible $O(n!)$.



Tipos de complejidad de un algoritmo

- 1 Resolver una ecuación cuadrática $O(1)$
- 2 Búsqueda lineal $O(n)$.
- 3 Recorrido de una matriz $O(n^2)$
- 4 Búsqueda binaria $O(\log(n))$
- 5 Agente viajero : visitar n países en el menor tiempo posible $O(n!)$.



Introducción

La eficiencia del algoritmo

Un algoritmo es eficiente cuando logra dar solución a un problema y logra optimizar los recursos disponibles (tiempo y espacio de memoria).

Utilidad de un algoritmo eficiente

- 1 En dispositivos con poca disponibilidad de memoria.
- 2 Cuando hay gran cantidad de datos a procesar.



Introducción

La eficiencia del algoritmo

Un algoritmo es eficiente cuando logra dar solución a un problema y logra optimizar los recursos disponibles (tiempo y espacio de memoria).

Utilidad de un algoritmo eficiente

- 1 En dispositivos con poca disponibilidad de memoria.
- 2 Cuando hay gran cantidad de datos a procesar.



Introducción



La Ciencia de la Computación incluye el estudio de cómo se organiza la información, cómo se maneja y cómo puede ser utilizada.

- 1 Un tipo de dato de una variable nos define el conjunto de valores que esta puede tomar. Por ejemplo entero, booleano, real, etc.
- 2 Un tipo de dato abstracto (TDA) es un modelo matemático compuesto por operaciones definidas sobre este modelo. Pueden ser lineales (TDA lista) o no lineales (TDA árbol o TDA grafo).

Ejemplo.

Por ejemplo en una universidad una lista de alumnos se debe tener **alumno** como elemento de la lista con los siguientes datos **código**, **especialidad**, **nombre**, **etc.** y sus operaciones pueden ser **insertar**, **buscar**, **eliminar**.

- 1 Un tipo de dato de una variable nos define el conjunto de valores que esta puede tomar. Por ejemplo entero, booleano, real, etc.
- 2 Un tipo de dato abstracto (TDA) es un modelo matemático compuesto por operaciones definidas sobre este modelo. Pueden ser lineales (TDA lista) o no lineales (TDA árbol o TDA grafo).

Ejemplo.

Por ejemplo en una universidad una lista de alumnos se debe tener **alumno** como elemento de la lista con los siguientes datos **código**, **especialidad**, **nombre**, etc. y sus operaciones pueden ser **insertar**, **buscar**, **eliminar**.

- 1 Un tipo de dato de una variable nos define el conjunto de valores que esta puede tomar. Por ejemplo entero, booleano, real, etc.
- 2 Un tipo de dato abstracto (TDA) es un modelo matemático compuesto por operaciones definidas sobre este modelo. Pueden ser lineales (TDA lista) o no lineales (TDA árbol o TDA grafo).

Ejemplo.

Por ejemplo en una universidad una lista de alumnos se debe tener **alumno** como elemento de la lista con los siguientes datos **código**, **especialidad**, **nombre**, **etc.** y sus operaciones pueden ser **insertar**, **buscar**, **eliminar**.

Una estructuras de datos es un conjunto de variables, quizá de distintos tipos, conectadas entre si de diversas formas. Se utilizan para implementar los TDA en un determinado lenguaje de programación. **Un mismo TDA puede ser implementado utilizando distintas estructuras de datos** y proveer la misma funcionalidad. La forma de definir la estructura de datos influye en la eficiencia de los algoritmos. Para implementar el TDA lista podemos usar

- Estructuras de Datos Estáticas (compuesta solo por variables estáticas)
- Estructuras de Datos Dinámicas (compuesta por una o más variables dinámicas o llamados también variables tipo puntero).

Una estructura de datos es un conjunto de variables, quizá de distintos tipos, conectadas entre sí de diversas formas. Se utilizan para implementar los TDA en un determinado lenguaje de programación. **Un mismo TDA puede ser implementado utilizando distintas estructuras de datos** y proveer la misma funcionalidad. La forma de definir la estructura de datos influye en la eficiencia de los algoritmos. Para implementar el TDA lista podemos usar

- Estructuras de Datos Estáticas (compuesta solo por variables estáticas)
- Estructuras de Datos Dinámicas (compuesta por una o más variables dinámicas o llamados también variables tipo puntero).

Una estructuras de datos es un conjunto de variables, quizá de distintos tipos, conectadas entre si de diversas formas. Se utilizan para implementar los TDA en un determinado lenguaje de programación. **Un mismo TDA puede ser implementado utilizando distintas estructuras de datos** y proveer la misma funcionalidad. La forma de definir la estructura de datos influye en la eficiencia de los algoritmos. Para implementar el TDA lista podemos usar

- Estructuras de Datos Estáticas (compuesta solo por variables estáticas)
- Estructuras de Datos Dinámicas (compuesta por una o más variables dinámicas o llamados también variables tipo puntero).

Una estructuras de datos es un conjunto de variables, quizá de distintos tipos, conectadas entre si de diversas formas. Se utilizan para implementar los TDA en un determinado lenguaje de programación. **Un mismo TDA puede ser implementado utilizando distintas estructuras de datos** y proveer la misma funcionalidad. La forma de definir la estructura de datos influye en la eficiencia de los algoritmos. Para implementar el TDA lista podemos usar

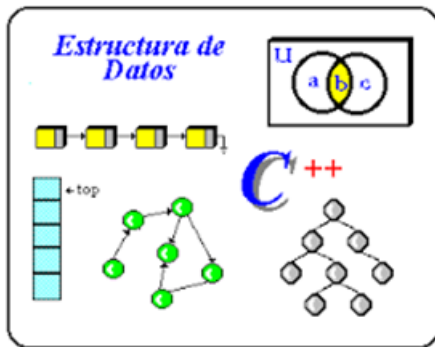
- Estructuras de Datos Estáticas (compuesta solo por variables estáticas)
- Estructuras de Datos Dinámicas (compuesta por una o más variables dinámicas o llamados también variables tipo puntero).

Una estructuras de datos es un conjunto de variables, quizá de distintos tipos, conectadas entre si de diversas formas. Se utilizan para implementar los TDA en un determinado lenguaje de programación. **Un mismo TDA puede ser implementado utilizando distintas estructuras de datos** y proveer la misma funcionalidad. La forma de definir la estructura de datos influye en la eficiencia de los algoritmos. Para implementar el TDA lista podemos usar

- Estructuras de Datos Estáticas (compuesta solo por variables estáticas)
- Estructuras de Datos Dinámicas (compuesta por una o más variables dinámicas o llamados también variables tipo puntero).

Introducción

Son estructura de datos un arreglo o vector, una matriz, un registro o estructura, una lista enlazada, una pila, una cola, un árbol, un grafo, etc.



Los peores errores de programación

Observe estas publicaciones y otras similares, y comparta sus contenidos.

1. <https://actualidad.rt.com/actualidad/view/138158-catastrofes-programacion-culpa-software-computadora>
2. https://www.1000tipsinformaticos.com/2017/11/los-10-peores-errores-de-programacion-en-la-historia.html?fbclid=IwAR0rnsydgHEqblfXUurlzIsvP_QqFWsYNwAhXRBTZnBlgpgNTh3CI – SAzFc