

Universidad Nacional de Asunción

Facultad Politécnica



Informe: Comparación de Lenguajes

Alumno: Diego Ramón Noguera Areco

Carrera: Ingeniería en Informática

Asignatura: Estructura de los Lenguajes

Año 2024

Problema 1: Ordenamiento sobre registros.

En todos los lenguajes, menos en R, se utilizan clases para guardar los datos. En R se utilizan data frames, que es una estructura de datos, similar a las estructuras en C.

En todos los lenguajes está escrita la función `quick_sort_calificacion` (o similar) que es la que se encarga de hacer el ordenamiento.

El algoritmo consiste en seleccionar un elemento del array al que se le llama pivote, y a partir de eso formar 2 nuevas listas. Una de las listas contiene a todos los elementos menores que el pivote, la otra contiene a todos los elementos mayores que el pivote. Una vez se tengan estas listas se las procede a ordenar recursivamente. El caso base para esta función recursiva es cuando la lista tiene 0 o 1 elementos, lo que quiere decir que está ordenada.

Un punto importante aquí es la selección del pivote. Gran parte de la eficiencia de este algoritmo depende de la selección adecuada del pivote. Para este caso se usa el método de la mediana de tres, como describe Clifford A. Shaffer en su libro *Data Structures and Algorithm Analysis*.

One good choice is to use the “median of three” algorithm, which uses as a pivot the middle of three randomly selected values. Using a random number generator to choose the positions is relatively expensive, so a common compromise is to look at the first, middle, and last positions of the current subarray. (Shaffer, 2011)

Comparación entre Lenguajes

Python se destaca por su sintaxis simple y directa. Se utiliza indentación para definir bloques de código, lo que mejora la legibilidad y elimina la necesidad de utilizar llaves como delimitadores de los bloques de código. Ruby al igual que Python tiene sintaxis bastante sencilla y descriptiva. Una diferencia es que en Ruby se utiliza `END` para delimitar los bloques de código, lo cual puede afectar negativamente la legibilidad. R tiene una sintaxis más especializada y orientada a manipulación de datos (por ejemplo los data frames, que son usados en el código). C# y Java tienen sintaxis similares, basados en objetos. Requieren más líneas de código debido a la necesidad de declarar explícitamente los tipos y estructuras. Esto causa que su código en general sea menos legible que el de Python o Ruby.

Python es super flexible para manejar listas y objetos. Escribir código para trabajar con colecciones es rápido y eficiente. Ruby sigue un camino similar, incluso permite mayor flexibilidad en la manipulación. R tiene estructuras para trabajar con grandes cantidades de datos como el ya mencionado data frame, que es muy poderosa para manejar datos tabulares aunque con una sintaxis

no tan sencilla. C# y Java de vuelta requieren más trabajo para manejar la estructura ya que son de tipado fuerte y orientado a objetos.

En cuanto a la implementación del algoritmo, Python permite una implementación directa y limpia gracias a su sintaxis sencilla y su capacidad para trabajar con listas de manera eficiente. Ruby tiene un comportamiento similar en este sentido. En R implementar el algoritmo resulta más complejo debido al manejo de data frames. C# y Java requieren más líneas de código y esfuerzo para gestionar las clases y objetos involucrados, pero a cambio proporcionan una estructura más clara y controlada.

Problema 2: Búsqueda binaria en estructuras de datos conglomeradas

Este problema consiste en realizar una búsqueda binaria sobre una lista de registros (alumnos) que contiene información sobre el nombre, edad y calificación. La búsqueda binaria se lleva a cabo utilizando la calificación como criterio de búsqueda.

La búsqueda binaria divide de forma iterativa la lista de elementos en dos partes. Cada vez, compara el valor buscado con el valor medio de la lista. Si el valor se encuentra en el medio, se devuelve el índice. Si no, se ajusta el intervalo de búsqueda para que incluya solo los elementos que podrían contener el valor buscado. Este proceso se repite hasta que se encuentra el valor o se determina que no está presente en la lista.

Comparación entre lenguajes

Python sobresale por su sintaxis simple y clara. Al usar indentación para definir bloques de código, se elimina la necesidad de llaves, lo que contribuye a un código más limpio y fácil de leer. Ruby tiene una sintaxis similar, pero utiliza la palabra END para cerrar bloques, lo que puede restar legibilidad en comparación con Python. R tiene una sintaxis más especializada para trabajar con datos tabulares mediante data frames. Si bien R es potente en análisis de datos, su sintaxis no es tan directa o intuitiva para implementar algoritmos como la búsqueda binaria, especialmente cuando se compara con lenguajes como Python. Java y C# comparten una sintaxis más formal y verbosa debido a su enfoque orientado a objetos. Ambos lenguajes requieren la declaración explícita de tipos y estructuras, lo que hace que el código sea más largo pero más estructurado. Esto puede dificultar la legibilidad en comparación con Python y Ruby, pero proporciona un control más estricto sobre los tipos de datos y las operaciones.

Python y Ruby son altamente eficientes para trabajar con listas y estructuras dinámicas. Esto se debe a sus poderosas capacidades de manejo de colecciones y su flexibilidad al trabajar con datos no tipados. Python sobresale con su eficiencia y facilidad al trabajar con listas y otros tipos de colecciones, lo que facilita implementar algoritmos como la búsqueda binaria.

R es excelente para trabajar con grandes volúmenes de datos estructurados en data frames, pero es más adecuado para tareas estadísticas y de análisis de datos. La implementación de una búsqueda binaria en R requiere un enfoque algo más complejo debido a la necesidad de convertir o adaptar la estructura de datos a la búsqueda binaria.

C# y Java requieren un poco más de esfuerzo debido a su tipado fuerte y la necesidad de gestionar explícitamente las estructuras de objetos. Sin embargo, sus sistemas de tipo ayudan a evitar errores en tiempo de compilación y hacen que el código sea más seguro y robusto, lo cual es útil en aplicaciones grandes y complejas.

Python y Ruby permiten una implementación directa y concisa de la búsqueda binaria. La estructura del código es simple y fácil de entender, lo que facilita la implementación de algoritmos clásicos como la búsqueda binaria. R complica un poco la implementación debido a la utilización de data frames. Aunque R es extremadamente eficiente para la manipulación de datos en masa, su sintaxis especializada no es la más conveniente para implementar este tipo de algoritmos. C# y Java requieren más líneas de código y una estructura más detallada debido a sus paradigmas orientados a objetos. Sin embargo, esto también les da una gran ventaja en términos de organización y control sobre los datos y las operaciones que realizan.

Problema 3: Multiplicación de matrices

Este problema consiste en realizar la multiplicación de matrices de toda la vida, fila por columna.

Comparación entre lenguajes

Python tiene sintaxis simple y directa, lo que hace que el código sea fácil de leer y entender. En este lenguaje, las listas se utilizan para representar las matrices y no es necesario declarar explícitamente los tipos de las variables, lo que mejora la legibilidad del código. La capacidad de Python para manejar colecciones como listas de manera flexible permite una implementación rápida del algoritmo de multiplicación de matrices sin requerir complejidad adicional.

Ruby, al igual que Python, tiene una sintaxis sencilla y clara. Es similar a Python en términos de manejo de colecciones y listas, pero utiliza END para marcar el final de bloques de código, lo que

puede restar flexibilidad frente a la indentación automática de Python. Aunque Ruby ofrece una implementación comparable a la de Python, su uso de END puede afectar la claridad en programas más grandes.

R es un lenguaje orientado a la manipulación de datos, especialmente en el análisis estadístico. En R, las matrices son un tipo de dato nativo y cuentan con muchas funciones predefinidas para realizar operaciones con ellas. Sin embargo, la implementación de la multiplicación de matrices en R es algo más compleja debido a que este lenguaje está diseñado principalmente para manejar grandes volúmenes de datos, lo que requiere más pasos en comparación con lenguajes como Python y Ruby.

Java y C# son lenguajes orientados a objetos que requieren una mayor cantidad de código para manejar las operaciones de matrices. Ambos requieren declarar explícitamente los tipos de las matrices y otros elementos, lo que hace que la implementación sea más verbosa. Aunque Java y C# proporcionan una estructura más rígida que ayuda a detectar errores en tiempo de compilación, esta necesidad de especificar tipos puede hacer que el código sea más largo y menos flexible en comparación con los lenguajes dinámicos.

Conclusión

Entre los lenguajes usados, Python es, en mi opinión, el más interesante. Su sintaxis simple, la flexibilidad para manejar colecciones y la gran cantidad de librerías disponibles lo convierten en una opción ideal para tareas matemáticas y científicas. Además, su popularidad en áreas como la ciencia de datos y la inteligencia artificial, sumada a su fácil curva de aprendizaje, lo hacen una herramienta poderosa y accesible para desarrolladores de todo tipo.

Sin embargo, tanto Ruby como Java y C# tienen su valor dependiendo del contexto en el que se utilicen. Ruby es excelente para desarrollo web y proyectos pequeños, mientras que Java y C# son más adecuados para aplicaciones empresariales o sistemas más complejos. En resumen, la elección del lenguaje depende de las necesidades específicas del proyecto, pero en términos de rapidez, flexibilidad y eficiencia para implementar algoritmos como la multiplicación de matrices, Python sobresale como la opción más interesante y práctica, para todos estos problemas presentados.

Fuentes

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*.

McGraw-Hill.

Sebesta, R. W. (n.d.). *Concepts of Programming Languages*. Pearson.

Shaffer, C. A. (2011). *Data Structures & Algorithm Analysis in Java*. Dover Publications.