

Gestión de sentencias: Proyecto de programación

Daniel Franco Barranco
Iván Matellanes Pastoriza
Asier Mujika Aramendia

Estructuras de Datos y Algoritmos
Grado en Ingeniería Informática

Diciembre de 2012

Índice

Resumen	3
Introducción.....	3
Versión 0.....	3
Versión 1.....	3
Actas de las reuniones.....	4
Anexos.....	4

Resumen

Aquí debe presentarse un resumen de todo el trabajo, que no debe superar la media página (más o menos). Debe explicarse en qué consiste el trabajo presentado, qué partes tiene y de qué trata cada parte.

Introducción

En esta sección debe explicarse el escenario del trabajo. Se presentarán también los objetivos del mismo: enunciado del problema a resolver y el alcance de funcionalidad de la implementación que se ha desarrollado.

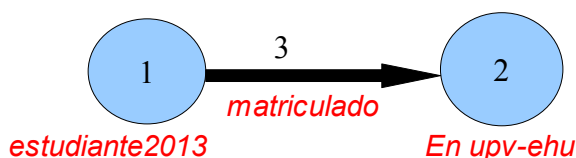
Puede ser interesante explicar el historial de las dificultades relevantes que se han superado.

En esta asignatura se nos presenta el problema de gestionar sentencias simples que tendrá la forma de sujeto propiedad objeto. (La idea principal).

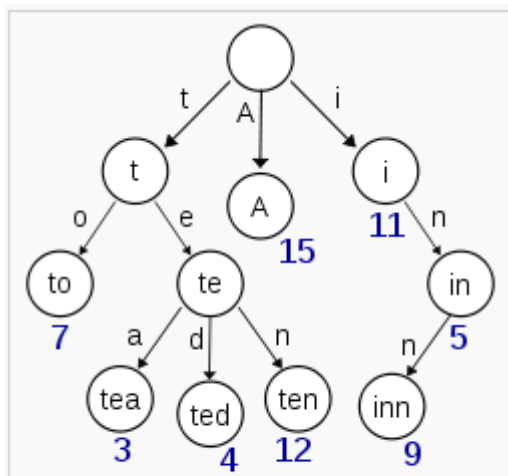
Versión 0

Para abordar este proyecto decidimos un par de ideas para estructurar los métodos. La idea fue esta: Ordenar todas las sentencias utilizando un grafo dirigido en el que los nodos serían los sujetos y objetos, y las aristas serían las propiedades de las sentencias. Asignándole a cada objeto, sujeto y propiedad un valor numérico para posteriormente administrarlo todo con una matriz de adyacencia. Para relacionar el valor numérico con el sujeto, objeto o propiedad correspondiente utilizaremos un árbol de prefijos (trie) y relacionar este con un array. Una vez tengamos estos métodos creados, y la matriz de adyacencia implementada procederemos a la creación de las funciones que se nos pide de una manera rápida, sencilla y eficiente.

1º paso:



2º paso:



Un trie de las claves "A", "to", "tea", "ted", "ten", "in", and "inn".

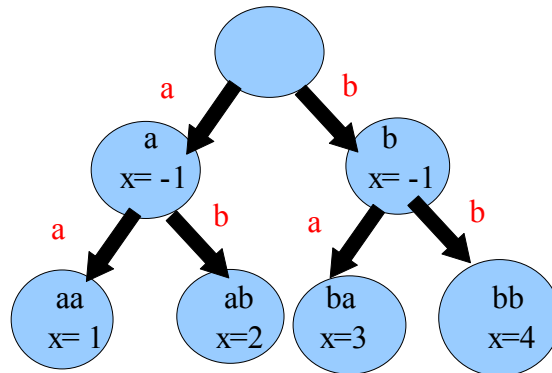
La idea es esta. Como en el ejemplo se puede observar para construir una palabra recorreremos el árbol hasta completar la palabra deseada. En nuestro caso, haremos el mismo árbol, guardando sus características pero con un pequeño cambio para relacionar las palabras con los números. Se añadirá a cada nodo aparte de la concatenación de caracteres que le corresponde, un valor numérico. Si la palabra estuviese en el array ese valor (variable X) tomaría el índice de la posición en la que se encuentra esa palabra, por el contrario tendrá el valor "-1" por defecto.

Ejemplo 2º paso:

Las palabras que están en el array son estas y se representarían así:

ARRAY:

1	2	3	4	
aa	ab	ba	bb



.....

Puesto que se trata de dejar constancia de todo el trabajo realizado durante el curso, es interesante plasmar en distintas secciones los estados principales por los que ha ido pasando la aplicación desarrollada. Así, podemos redactar una sección, que corresponderá con una nueva versión, por cada organización fundamental de los datos a procesar (diseño de datos) y sus correspondientes implementaciones de métodos que hayamos alcanzado con esa versión.

Se explicarán brevemente las decisiones de diseño de las estructuras de datos seleccionadas para resolver los problemas que se aborden inicialmente y se enumerarán los métodos que se han implementado.

Para cada método se indicará lo que hace (especificación) y, si es relevante, se dará también una breve explicación de cómo lo hace. Cada método irá acompañado de su clasificación en complejidad algorítmica y de una justificación de esa clasificación. Además, para los métodos más importantes, se realizarán pruebas de ejecución sobre datos de entrada de distintos tamaños para registrar el tiempo necesitado por esos métodos y se presentarán las tablas de resultados del modo más conveniente para su interpretación.

El código fuente de los programas implementados puede agruparse en un anexo que se presenta en una sección creada al efecto al final de la memoria.

Versión 1

Según vayamos avanzando en el curso y vayamos conociendo nuevas formas de estructurar los datos y nuevos modos de tratar esas estructuras, descubriremos que la

Versión 0, realizada como primera opción, es mejorable con la aplicación de esos nuevos conocimientos.

Aprovecharemos todo lo que se pueda de la versión anterior (y no habrá que presentarlo en esta nueva sección, bastará con indicarlo). Con cada nueva versión, se explicarán y se justificarán los cambios fundamentales en las estructuras de datos y los métodos correspondientes.

Se mantendrá el estilo de la presentación: descripción y justificación de la representación de los datos, enumeración y especificación de cada método, análisis de eficiencia y tablas de tiempos de ejecución de los más relevantes.

Versión 2

Es más que probable que convenga modificar las estructuras de datos más de una vez, ya que vamos a ir descubriendo estructuras interesantes durante todo el curso. Si se encuentra algo mejor, se valorará su incorporación a la aplicación. La finalidad del proyecto de programación es demostrar que se conocen las distintas estructuras de datos que iremos viendo en el curso y que se saben usar allá donde conviene para conseguir aplicaciones eficientes.

En cada sección, dedicada al efecto, se explicarán las razones de la correspondiente modificación. El código fuente se presentará, agrupado, en la correspondiente sección de anexos.

Actas de las reuniones

01/10/2012:

En esta primera reunión hemos decidido varios aspectos a tener en cuenta y nos hemos repartido la tarea en varias partes:

- Hemos terminado de debatir las estructuras que vamos a utilizar en el proyecto.
- A cada miembro del grupo se les ha asignado un tipo de tarea que realizar:
 - Asier Mujika Aramendia realizará la implementación de el árbol necesario para la creación del array.
 - Iván Matellanes Pastoriza realizará las funciones respecto al uso del fichero.
 - Daniel Franco Barranco probará casos de prueba para las funciones creadas anteriormente por Iván y realizará la parte de la memoria correspondiente a los primeros pasos del proyecto.

Conclusión del acta: Tras un día de reparto de los trabajos individuales se han completado con éxito todas las expectativas. En la siguiente reunión grupal se decidirán los nuevos trabajos a cada miembro del grupo.

En esta sección se irán redactando, en orden cronológico y “en tiempo real”, las actas de las reuniones del grupo de trabajo. La periodicidad y el número de reuniones será una decisión del grupo de trabajo; pero parece razonable que, al menos, haya una

cada semana. Cada acta llevará la fecha de realización y duración de la reunión. Cada acta tendrá una redacción breve; me extrañaría que fuese necesario superar la media página.

En estas actas deben reflejarse las decisiones tomadas en cada reunión y deben reflejarse los hitos relevantes del proyecto. Por ejemplo, deberá reflejarse el reparto de tareas entre las personas del grupo decidido para un objetivo concreto y, cuando sea el caso, reflejará los objetivos que se han conseguido al terminar determinada tarea. De este modo se puede tener un seguimiento del proceso de desarrollo del proyecto. Para ello es primordial que esas actas se redacten en el momento correspondiente y no se demore nunca su redacción más allá del comienzo de la siguiente reunión.

Las decisiones de reparto de las tareas pueden ser fundamentales para un eficiente desarrollo del proyecto. Por ejemplo, mientras alguien realiza la implementación de unos métodos, otra persona puede realizar los métodos de prueba de los mismos y una tercera redactar las secciones correspondientes para la memoria. Puede ser interesante que, después, cada cual revise algún aspecto crítico de la tarea realizada por otro miembro del grupo, para asegurar la corrección de la misma. Este modo de actuar puede ser mucho más eficiente que el intento de realizar todos todo y a la vez. El modo de trabajo se verá reflejado en las actas de las reuniones.

Anexos

Anexo versión 0

Código fuente de lo implementado para la versión 0.

Anexo versión 1

Código fuente de lo implementado para la versión 1.

Anexo versión 2

Código fuente de lo implementado para la versión 2.