Proyecto Final

Fundamentos y Estructuras de Programación

Departamento de Ingeniería de Sistemas
Pontificia Universidad Javeriana

Profesor Carlos Alberto Ramírez Restrepo
carlosalbertoramirez@javerianacali.edu.co

El presente proyecto tiene como objetivo enfrentar a los estudiantes del curso:

- Al diseño e implementación de Tipos Abstractos de Datos.
- Al análisis de la complejidad computacional de la implementación de operaciones primitivas de un Tipo Abstractos de Datos.
- Al análisis y comparación en términos de complejidad computacional de diferentes estrategias de representación para un mismo Tipo Abstracto de Datos.
- Al diseño e implementación de operaciones generales sobre un Tipo Abstracto de Datos.

1 Matrices Dispersas

Una *matriz dispersa* corresponde a un tipo de matriz de gran tamaño en la cuál la cantidad de información relevante es baja en comparación con las dimensiones de la matriz. Típicamente, en este tipo de matrices la información no relevante se representa por el valor 0 (cero) y la información relevante por valores diferentes a 0. De esta manera, en este tipo de matrices es posible predecir que el porcentaje de ceros es alto. En general, considerar una matriz dispersa, osea, si la cantidad de ceros es lo suficientemente alta, depende del contexto específico en el cuál se trabaje.

Las matrices son estructuras de datos que en general son muy eficientes en cuanto al uso de memoria y de procesador. Sin embargo, la representación computacional de las matrices dispersas mediante el enfoque general tiene como consecuencia un uso ineficiente de la memoria y del tiempo de procesamiento. Esto se debe a que se utilizan muchas posiciones de memoria para almacenar ceros y se requiere procesar todas las posiciones de la matriz innecesariamente.

En consecuencia, teniendo como objetivo mejorar la complejidad espacial y temporal de las operaciones y de la implementación específica de matrices dispersas, se han propuesto diversas representaciones. De esta manera, las matrices dispersas pueden ser definidas

0	2	0	0	0	0	4
0	8	9	0	0	1	0
0	0	0	3	0	0	0
0	0	0	0	0	0	0
5	0	0	0	0	6	0
1	2	0	0	0	0	0
4	0	0	0	0	0	0
0	0	7	0	0	11	0

Table 1: Ejemplo Matriz Dispersa

y especificadas como un TAD. El propósito de dichas representaciones es almacenar únicamente la información relevante de cada fila y columna de la matriz original con el propósito de utilizar menos memoria y agilizar las operaciones. En la siguiente sección se describen los principales enfoques de representación de matrices dispersas.

2 Representaciones

En las siguientes secciones se describirán los enfoques de representación más comúnes para matrices dispersas. En adelante, se utilizará `n` y `m` para denotar el número de filas y columnas de la matriz completa, en tanto que se utilizará `ne` para denotar el número de elementos diferentes de cero. Además, cada una de las representaciones será ilustrada utilizando como referencia la matriz dispersa de la Tabla 1.

2.1 Formato Coordenado

La representación de matrices dispersas mediante el formato coordenado almacena los datos distintos de cero de la matriz dispersa junto con su ubicación en la matriz, i.e., el índice de su fila y su columna. Para esto se utilizan 3 vectores de tamaño `ne`. En el primer vector se almacen los valores distintos de cero de la matriz original mientras que en el segundo y tercer vector se almacenan los índices de la fila y la columna para cada uno de los valores distintos de cero.

Los siguientes serían los valores en los vectores para la representación en formato coordenado de la matriz dispersa de la Tabla 1:

```
valores = [2 4 8 9 1 3 5 6 1 2 4 7 11]
  filas = [0 0 1 1 1 2 4 4 5 5 6 7 7]
columnas = [1 6 1 2 5 3 0 5 0 1 0 2 5]
```

2.2 Formato Comprimido

Compressed Sparse Row: El formato comprimido por fila (CSR) representa una matriz dispersa por medio de 3 vectores. El primer vector tiene tamaño `ne` y almacena los valores distintos de cero de la matriz original organizados fila por fila. En el segundo vector, que también tiene tamaño `ne`, se almacenan los índices de las columnas en las que están cada uno de los valores del primer vector en la matriz original. En tanto que, el tercer vector tiene tamaño `n`+1 y almacena la posición donde empiezan los valores de cada fila en el segundo vector (vector de columnas).

Los siguientes serían los valores en los vectores para la representación en el formato CSR para la matriz de la Tabla 1:

```
valores = [2 4 8 9 1 3 5 6 1 2 4 7 11]
   cols = [1 6 1 2 5 3 0 5 0 1 0 2 5]
  cfilas = [0 2 5 6 6 8 10 11 13]
```

Observe que el tercer vector (vector `cfilas`) se puede interpretar asociando el valor c_i y el valor $c_{i+1} - 1$ como el índice inicial y el índice final en el vector de columnas asociados a la fila i. De esta manera, para el ejemplo anterior se tiene:

Fila	Límites	Columnas asociadas
0	cfilas[0] a cfilas[1]-1	cols[0], cols[1]
1	cfilas[1] a cfilas[2]-1	cols[2], cols[3], cols[4]
2	cfilas[2] a cfilas[3]-1	cols[5]
3	cfilas[3] a cfilas[4]-1	-
4	cfilas[4] a cfilas[5]-1	cols[6], cols[7]
5	cfilas[5] a cfilas[6]-1	cols[8], cols[9]
6	cfilas[6] a cfilas[7]-1	cols[10]
7	cfilas[7] a cfilas[8]-1	cols[11], cols[12]

Es importante resaltar que en esta representación cuando una fila no contiene ningún valor distinto a cero entonces el valor asociado a dicha fila en el tercer vector (vector `cfilas`) es el mismo de la fila anterior. Además, el último valor en el tercer vector corresponde al último índice válido en el segundo vector más 1.

Compressed Sparse Column: El formato comprimido por columna (CSC) también representa una matriz dispersa por medio de 3 vectores. Este formato es análogo al formato CSR y utiliza también 3 vectores. El primer vector tiene tamaño `ne` y almacena los valores no nulos en la matriz original organizados por columna. El segundo vector tiene tamaño `ne` y almacen los índices de las filas en las que están cada uno de los valores del primer vector. Mientras que el tercer vector tiene tamaño `m`+1 y almacena la posición donde empiezan los valores de cada columna en el segundo vector (vector de filas).

Los siguientes serían los valores en los vectores para la representación en el formato CSC para la matriz de la Tabla 1:

```
valores = [5 1 4 2 8 2 9 7 3 1 6 11 4]
filas = [4 5 6 0 1 5 1 7 2 1 4 7 0]
ccolumnas = [0 3 6 8 9 9 12 13]
```

Observe que el tercer vector (vector `ccolumnas`) se puede interpretar asociando el valor c_i y el valor $c_{i+1} - 1$ como el índice inicial y el índice final en el vector de filas asociados a la columna i. De esta manera, para el ejemplo anterior se tiene:

Fila	Límites	Columnas asociadas
0	ccolumnas[0] a ccolumnas[1]-1	filas[0], filas[1] filas[2]
1	ccolumnas[1] a ccolumnas[2]-1	filas[3], filas[4], filas[5]
2	ccolumnas[2] a ccolumnas[3]-1	filas[6], filas[7]
3	ccolumnas[3] a ccolumnas[4]-1	filas[8]
4	ccolumnas[4] a ccolumnas[5]-1	-
5	ccolumnas[5] a ccolumnas[6]-1	filas[9], filas[10] filas[11]
6	ccolumnas[6] a ccolumnas[7]-1	filas[12]

Como en la representación CSR, cuando una columna no contiene ningún valor distinto a cero entonces el valor asociado a dicha columna en el tercer vector (vector `ccolumnas`) es el mismo de la columna anterior. Además, el último valor en el tercer vector corresponde al último índice válido en el segundo vector más 1.

2.3 Listas Enlazadas por Fila

Una matriz dispersa puede ser representada mediante listas enlazadas. De esta manera, en esta representación se tiene una lista o vector de listas enlazadas. Cada una de las listas enalazadas representa los elementos en una fila o en una columna de la matriz original.

La Figura 1 muestra la representación mediante listas enlazadas de la matriz dispersa de la Tabla 1. En la figura se asume que se utilizan listas doblemente enlazadas aunque también es posible utilizar listas simples o listas circulares. Observe que las filas que no tiene valores distintos a cero son representadas mediante listas vacías. En cada nodo de la lista se almacena un valor distinto de cero de la matriz original y el índice de la columna en donde está ubicado dicho valor. Finalmente, es posible construir una representación análoga en la que cada lista enlazada esté asociada a cada columna en la matriz original.

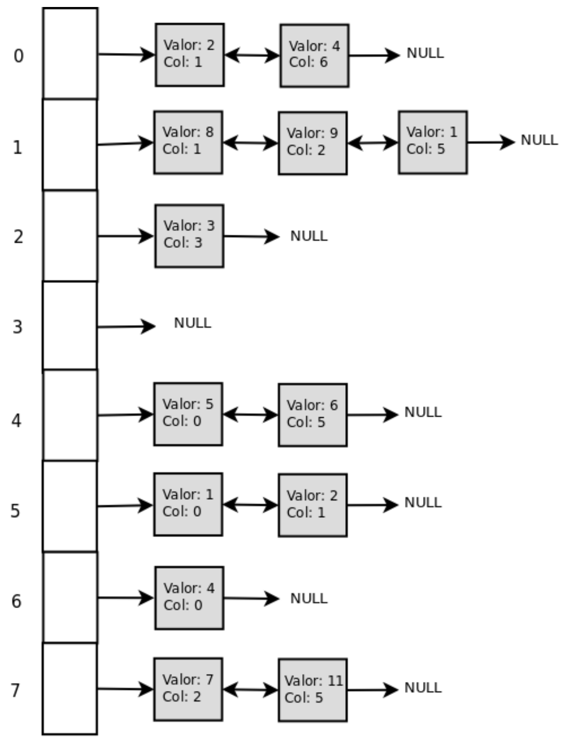

Figure 1: Representación Listas Enlazadas Matriz Tabla 1

3 Operaciones

A continuación se describe brevemente las operaciones básicas internas para el TAD matriz dispersa:

Crear de Matriz Completa: Esta operación permite crear una instancia de una matriz dispersa a partir de una matriz completa (típicamente representada como un arreglo de dos dimensiones).

Obtener Matriz Completa: Esta operación construye y retorna la matriz completa asociada a la instancia de una matriz dispersa.

Obtener Elemento: Esta operación permite obtener el elemento en la posición i, j en la matriz dispersa. Si la posición i, j corresponde a un cero, este valor debe ser retornado.

Obtener Fila: Esta operación permite obtener la fila j de la matriz dispersa como un vector o como una lista enlazada (se deben considerar ambas variantes).

Obtener Columna: Esta operación permite obtener la columna j de la matriz dispersa como un vector o como una lista enlazada (se deben considerar ambas variantes).

Obtener Fila Dispersa: Esta operación permite obtener la fila j de la matriz dispersa incluyendo los ceros como un vector o como una lista enlazada (se deben considerar ambas variantes).

Obtener Columna Dispersa: Esta operación permite obtener la columna j de la matriz dispersa incluyendo los ceros como un vector o como una lista enlazada (se deben considerar ambas variantes).

Obtener número de elementos: Esta operación permite obtener el valor `ne` de una matriz disperasa, i.e., el número de valores distintos de cero.

Modificar Posición: Esta operación permite modificar el valor en la posición i, j de la matriz. Es importante resaltar que si el valor anterior es un cero en la matriz original o si el nuevo valor es un cero, se debe modificar la estructura de la matriz dispersa para incluir el nuevo valor o para eliminar la posición que pasa a contener un cero.

Por otro lado, aunque las matrices dispersas pueden ser utilizadas de la misma manera que las matrices no dispersas, en este proyecto se hará énfasis en las siguientes operaciones:

Suma de Matrices: En esta operación se reciben dos matrices dispersas A y B de iguales dimensiones y se retorna una nueva matriz dispersa asociada a la suma de A y B.

Producto Matriz Vector: En esta operación se recibe una matriz dispersa y un vector (representado como un arreglo o como una lista enlazada) y retorna el resultado de multiplicar la matriz por el vector.

Matriz Transpuesta: En esta operación se recibe una matriz dispersa y se debe construir la transpuesta de dicha matriz.

4 Entrega

En el presente proyecto se espera que usted implemente tres representaciones del TAD matriz dispersa. Más específicamente, cada grupo de trabajo debe:

1. Implementar el TAD matriz dispersa con la representación en el formato coordenado, en uno de los formatos comprimidos y en el formato de listas enlazadas por fila o columna.
2. Incluir en cada implementación todas las operaciones descritas en este documento.
3. Analizar la complejidad de cada operación. No es necesario que se analice detalladamente cada linea de cada operación pero se debe explicar claramente la complejidad que se reporte para cada una de ellas.
4. Hacer una comparación entre las tres implementaciones teniendo en cuenta la complejidad de las operaciones y los aspectos generales de la implementación.
5. Explicar claramente las decisiones de implementación realizadas y probar cada una de las operaciones implementadas.

5 Informe

El informe debe entregarse en un archivo **pdf** y debe contener las explicaciones de las diferentes decisiones que se tomen en la implementación y las conclusiones del proyecto. Más específicamente, el informe debe contener:

- Detalles principales de las tres implementaciones de matrices dispersas.
- Análisis de la complejidad de estas implementaciones.
- Conclusiones y aspectos a mejorar. Siendo esta una de las partes más interesantes del trabajo, usted debe **analizar los resultados obtenidos** y **justificar** cada una de sus afirmaciones.

Aclaraciones

1. El proyecto se debe realizar en **grupos de 3** personas. Cualquier indicio de copia causará la anulación del proyecto y la ejecución del proceso disciplinario que corresponda de acuerdo a la normativa de la universidad.
2. Para la entrega debe generar un archivo `.zip` o `.tar.gz` con el contenido del proyecto (código fuente, documentación del proyecto y ayudas) el cual debe seguir la convención *Apellido1Apellido2Apellido3ProyectoFEP17.zip*. Deben incluir un archivo llamado `informe.pdf`, correspondiente al informe del proyecto.
3. La solución del proyecto debe ser enviada través de la plataforma *moodle* a más tardar el día **1 de Junio a las 9:00am**. Dependiendo de las fechas de corte del sistema es posible correr la entrega para el día **6 de Junio a las 9:00am** . Ese mismo día se llevará a cabo la sustentación del proyecto.

   La nota final del proyecto será individual y dependerá de la sustentación de cada estudiante. Cada estudiante, después de la sustentación tendrá asignado un número real (el factor de multiplicación) entre 0 y 1, correspondiente al grado de calidad de su sustentación. Su nota definitiva será la nota del proyecto, multiplicada por ese valor. Si su asignación es 1, su nota será la del proyecto. Pero si su asignación es 0.9, su nota será 0.9 por la nota del proyecto. La no asistencia a la sustentación tendrá como resultado una asignación de un factor de 0.
4. Recuerde, en caso de dudas y aclaraciones puede preguntar al inicio de las clases o enviar un correo con su consulta al profesor.