**TAD Matriz Dispersa**

**SEBASTIAN IZQUIERDO SAAVEDRA**

**JOSHUA EMMANUEL MENDEZ**

**12/11/2023**

**ESTRUCTURA DE DATOS**

**PROFESOR: CARLOS RAMIREZ**

**MATRIZ DISPERSA**

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 si una matriz es dispersa o no, osea, si la cantidad de ceros en la matriz es lo suficientemente alta, depende del contexto específico en el cuál se trabaje.

Para este **TAD Matriz Dispersa** se usaron dos implementaciones las cuales fueron **Formato Coordenado** y **Listas Enlazadas por Fila**.

**Informe de Complejidades**

1. DisperseMatrix::DisperseMatrix(int \*\*&*matriz*, int *fil*, int *col*)

Una operación constructora que tome como parámetro una matriz completa representada como un arreglo de dos dimensiones.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n\*m) | **Complejidad O(n\*m)** |
| Siendo n = el numero de filas y m = el numero de columnas | Siendo n = el numero de filas y m = el numero de columnas |

1. DisperseMatrix::DisperseMatrix( vector<vector<int>> &*matriz*)

Una operación constructora que tome como parámetro una matriz completa representada como un vector de vectores.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n\*m) | **Complejidad O(n\*m)** |
| Siendo n = el numero de filas y m = el numero de columnas | Siendo n = el numero de filas y m = el numero de columnas |

1. DisperseMatrix::DisperseMatrix(const DisperseMatrix &*matriz1*)

Una operación constructora que tome como parámetro otra instancia de DisperseMatrix y copie sus valores en el objeto que se está creando.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n)** |
| Siendo n = el tamaño del vector | Siendo n = el tamaño del vector de listas |

1. vector<vector<int>> DisperseMatrix::rebuild()

La operación rebuild que reconstruya la matriz completa asociada a la instancia de una matriz dispersa.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n\*m)** |
| Siendo n el tamaño del vector | Siendo n el tamaño del vector, y m el tamaño de la lista |

1. int DisperseMatrix::get(int *i*, int *j*)

La operación get que permita obtener el valor que está en la posición i, j en la matriz. Si la posición i, j corresponde a un cero, este valor debe ser retornado.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n)** |
| Siendo n el tamaño del vector | Siendo n el tamaño de la lista |

1. list<pair<int, int>> DisperseMatrix::getRowList(int *fila*)

La operación getRow que permite obtener la fila j de la matriz dispersa como una lista.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(1)** |
| Siendo n el tamaño del vector | Siendo n el tamaño de la lista |

1. vector<pair<int, int>> DisperseMatrix::getRowVec(int *fila*)

La operación getRow que permite obtener la fila j de la matriz dispersa como un vector.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n)** |
| Siendo n el tamaño del vector | Siendo n el tamaño de la lista |

1. list<pair<int, int>> DisperseMatrix::getColList(int *columna*)

La operación getCol que permite obtener la columna j de la matriz dispersa como una lista.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n)** |
| Siendo n el tamaño del vector | Siendo n el tamaño de la lista |

1. vector<pair<int, int>> DisperseMatrix::getColVec(int *columna*

La operación getCol que permite obtener la columna j de la matriz dispersa como un vector.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n)** |
| Siendo n el tamaño del vector | Siendo n el tamaño de la lista |

1. vector<pair<int, int>> DisperseMatrix::getDisperseRowVec(int *fila*)

La operación getDisperseRow que permite obtener la fila j de la matriz dispersa incluyendo los ceros como un vector.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n)** |
| Siendo n el valor maximo entre nColumnas y el tamaño del vector de valores | Siendo n el valor maximo entre nColumnas y el tamaño de la lista |

1. list<pair<int, int>> DisperseMatrix::getDisperseRowList(int *fila*)

La operación getDisperseRow que permite obtener la fila j de la matriz dispersa incluyendo los ceros como una lista.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n)** |
| Siendo n el valor maximo entre nColumnas y el tamaño del vector de valores | Siendo n el valor maximo entre nColumnas y el tamaño de la lista |

1. vector<pair<int, int>> DisperseMatrix::getDisperseColVec(int *columna*)

La operación getDisperseCol que permite obtener la columna j de la matriz dispersa incluyendo los ceros como un vector.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n)** |
| siendo n = el valor maximo entre nFilas y el tamaño del vector de valores | Siendo n el valor maximo entre nFilas y el tamaño de la lista |

1. list<pair<int, int>> DisperseMatrix::getDisperseColList(int *columna*)

La operación getDisperseCol que permite obtener la columna j de la matriz dispersa incluyendo los ceros como una Lista.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n)** |
| siendo n = el valor maximo entre nFilas y el tamaño del vector de valores | Siendo n el valor maximo entre nFilas y el tamaño de la lista |

1. void DisperseMatrix::assign(int *i*, int *j*, int *v*)

La operación assign que 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.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n)** |
| Siendo n el tamaño del vector | Siendo n el tamaño de la lista |

1. void DisperseMatrix::add(DisperseMatrix &*matriz*)

La operación add que recibe otra matriz dispersa de iguales dimensiones y le suma al objeto actual la matriz que se recibe como parámetro.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n\*m) | **Complejidad O(n\*m\*x)** |
| Siendo n el tamaño del vector de la matriz 1 y m el tamaño del vector de la matriz 2 | Siendo n el tamaño de las filas de la matriz 1, m el tamaño de la lista de la matriz 1, x el tamaño de la lista de la matriz 2 |

1. void DisperseMatrix::printMatrix(string *sep*)

La operación printMatrix que recibe una cadena de separación e imprime cada fila de la matriz en una línea con los elementos separados por dicho caracter.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n\*m) | **Complejidad O(n\*m)** |
| Siendo n el numero de filas y m el numero de columnas | Siendo n el numero de filas y m el numero de columnas |

1. void DisperseMatrix::productVector(vector<int> &*vec*)

La operación productVector que recibe un vector y multiplica el objeto actual por el vector recibido.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n\*m\*x)** |
| Siendo n el tamaño del vector | Siendo n el tamaño de las columnas, m el tamaño de las filas y x el tamaño de la lista |

1. int DisperseMatrix::getMax()

La operación getMax que debe retornar el mayor elemento en la matriz.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n\*m)** |
| Siendo n el tamaño del vector | Siendo n el tamaño del vector de filas, y m el tamaño de las listas |

1. DisperseMatrix DisperseMatrix::getTranspose()

La operación getTranspose que construye la matriz dispersa correspondiente a la transpuesta del objeto actual.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n\*m)** |
| Siendo n el tamaño del vector | Siendo n el tamaño del vector de filas, y m el tamaño de la listas |

DisperseMatrix DisperseMatrix::addMatrixList(list<DisperseMatrix> &*listaMatriz*)

La operación estática addMatrixList que toman una lista de instancias del TAD DisperseMatrix y retorna el resultado de sumar la lista de matrices recibidas.

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n\*\*2) | **Complejidad O(n\*\*2\*m\*x)** |
| Siendo n el tamaño de la lista y la sobrecarga + que tiene costo de O(n) | Siendo n el tamaño de la lista y la sobrecarga + que tiene costo de O(n\*m\*x) |

**Sobrecarga de operador +**

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n\*m\*x)** |
| siendo n = el valor maximo entre el tamaño del vector valores de la matriz 1 y  la matriz 2 | Siendo n el tamaño de las filas de la matriz 1, m el tamaño de la lista de la matriz 1, x el tamaño de la lista de la matriz 2 |

**Sobrecarga de operador \***

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(m\*v\*n\*\*2) | **Complejidad O(m\*v\*n\*\*2)** |
| n = es el máximo entre nColumnas y matrix2.nFilas, m = nFilas, v = matriz2.nColumnas y get() que vale O(n) | n = es el máximo entre nColumnas y matrix2.nFilas, m = nFilas, v = matriz2.nColumnas y get() que vale O(n) |

**Sobrecarga de operador ==**

|  |  |
| --- | --- |
| Formato Coordenado | Listas Enlazadas por Fila |
| Complejidad O(n) | **Complejidad O(n\*m)** |
| n = tamaño del vector valores | n = número de filas y m = al tamaño máximo de la lista en la matriz 1 o la lista de la matriz 2 |

**Conclusión**

En conclusión, la implementación de un TAD Matriz Dispersa en el Formato Coordenado tiende a ofrecer mejoras en la complejidad en comparación con la representación de Listas Enlazadas por Fila.