Elecci\’on de estructuras

\indent Luego de haber estudiado los tres tipos de estructuras dados por la c\’atedra: Dictionary of Keys (DOK), Compressed Sparse Row (CSR) y Compressed Sparse Column (CSC), decidimos cual elegir. \\

\indent En primera instacia, consideramos la estructura DOK. La misma consiste en un diccionario con doble clave, donde cada una es fila y columna respectivamente y su significado son los elementos de la matriz distintos de cero. De esta manera, se aprovecha en t\’erminos de espacio en memoria la notable cantidad de ceros de la matriz. Contaba con la gran ventaja de que es buena para construirla incrementalmente en un arreglo esparso y adem\’as se puede trasponer de manera sencilla ya que es invertir el orden de las claves. En contraposici\’on, para procesar los c\’alculos aritm\’eticos es necesario convertirla a otro formato. Por este motivo, descartamos esta opci\’on. \\

\indent El modo de almacenamiento *Compressed sparse row* requiere la implementacion de tres arreglos (en nuestro caso vectores) que llamaremos val, ind\_col y ptr\_fila. El tamaño de los dos primeros estar\’a dado por la cantidad de elementos distintos de cero de la matriz. Mientras que el primero almacenar’a estos valores de izquierda a derecha y de arriba hacia abajo, el segundo vector guardar’a para cada posición de aquellos valores, el indice. Para cada indice del vector val, el mismo indice indica la columna de ese valor. Es decir, el elemento almacenado en la posición i-esima del vector ind\_col representa la columna correspondiente al valor almanacenado en val\_i. \\

\indent El modo de almancenamiento Compressed Sparse Column cuenta tambi\’en con la implementaci\’on de tres arreglos llamados: val, ind\_fila, ptr\_col. Val es un arreglo con todos los valores distintos de cero de la matriz, desde arriba hacia abajo y luego de izquierda a derecha. Ind\_fila son los indices de fila correspondientes a los valores. Por ultimo, ptr\_col lista los indices donde comienza cada columnas.\\

\indent En segunda instancia, nos encontramos frente a la disyuntiva sobre si elegir el formato CSR (por filas) o CSC (por columnas) ya que no notamos un beneficio de una sobre otra. \\

\indent Por ultimo, el vector ptr\_fila tendra de tamaño la cantidad de filas incrementada en uno y listara los indices que indicaran los valores de val que comienzan cada fila.\\

\indent DESPU\’ES DE ESO (?), haciendo c\’alculos pequeños notamos que si nos situamos en el formato de compressed spare trasponer una matriz almacenada de manera CSC no es m\’as que interpretar los mismos tres arreglos como CSC. VA DEMOSTRACION DE ESTO???. Fue decisi\’on del grupo considerar el formato por defecto de la matriz el CSR (filas) y al trasponerlas s\’olo modificarle un bool que indique si est\’a traspuesta y de ahora en mas leerla y considerarla como CSC (columnas). Esta decisión fue tomada luego de que Agust’in Montero nos confirmara que estaba permitido elegir una opción de las ofrecidas y adaptarla a nuestra conveniencia, siempre que se aclararan los cambios. Por este motivo, en el algoritmo de multiplicar una matriz por un vector diferencia la manera en que este almacenada y hace la multiplicaci\’on acorde a su manera respectiva, VER ALGORITMO EN ANEXO… \\