Alfred Jophy

CS27

Sparse Matrix Compression

1 . Array Method

```
//sparse matrix into linked list
#include <stdio.h>
#include <stdlib.h>

int* matrix_input(int *rows,int *columns,int *size){
        printf("Enter the number of rows and columns of the matrix\n");
        printf("Rows    : ");
        scanf("%d",rows);
        printf("Columns : ");
        scanf("%d",columns);

        *size=(*rows)*(*columns)*sizeof(int);
        int *matrix=calloc((*rows)*(*columns),sizeof(int));

        printf("Enter the elements of the matrix : \n");
        for(int i=0;i<*rows;i++)
                for(int j=0;j<*columns;j++){
                        scanf("%d",((matrix+i*(*columns))+j));
                }

        return matrix;
}

void matrix_display(int rows,int columns,int* matrix){
        for(int i=0;i<rows;i++){
                printf("\n");
                for(int j=0;j<columns;j++)
                        printf("%d ",*((matrix+i*columns)+j));
        }

        printf("\n");
}

//#########################################################
struct node{
        int x,y,val;
        struct node* link;
};
```

```
typedef struct node node;

node* prepend_list(node* start,int x,int y,int val){

        node* temp=(node *)calloc(1, sizeof(node));

        temp->link=start;
        start=temp;

        start->x=x;
        start->y=y;
        start->val=val;

        return start;
}

void display_list(node* start){
        for(node* i=start;i!=NULL;i=i->link){
                //printf stuff
                printf("%d %d %d\n",i->x,i->y,i->val);
        }
}

void free_list(node* start){
        while(start){
                node* temp=start;
                start=start->link;
                free(temp);
        }
}
int sizeof_list(node* start){
        int number_of_nodes=0;
        while(start){
                node* temp=start;
                start=start->link;
                number_of_nodes++;
        }
        return number_of_nodes*sizeof(node);
}


node* csm_linkedL_method(int rows,int columns,int *matrix){
        node* start=NULL;
        for(int i=0;i<rows;i++)
                for(int j=0;j<columns;j++)
                        if(*((matrix+i*columns)+j))
```

```
                                start=prepend_list(start,i,j,*((matrix+i*columns)+j));
        return start;
}

int main(){
        node* compressed_spare_matrix_list=NULL;

        int *matrix;
        int rows,columns;
        int size;

        printf("%lu",sizeof(struct node));

        matrix=matrix_input(&rows, &columns, &size);
        printf("The Matrix : \n");
        matrix_display(rows, columns, matrix);

        compressed_spare_matrix_list=csm_linkedL_method(rows, columns,matrix);
        printf("X Y Value\n");
        display_list(compressed_spare_matrix_list);

        printf("Size of Sparse Matrix(Bytes)            : %d\n",size);
        printf("Size of Compressed Linked List (Bytes) : %d\n",sizeof_list(compressed_spare_

        free(matrix);
        free_list(compressed_spare_matrix_list);

        return 0;
};
```