**Sparse\_matrix.c에 대한 instruction**

2017314875조규원

1.main 함수

1. 파일 입출력을 위해 FILE 포인터 변수인 rEad, wRite를 선언 후 fopen으로 파일 열기, 행렬 A, B를 입력 받기 위해 int형 더블 포인터 변수 a,b를 선언, 명령어를 입력 받기 위해 길이가 100인 char 형 배열 층 선언, 그 외 주어진 값 n,m, for문을 위한 변수 i,j를 선언

2. n값을 입력 받고 a와 b에 동적할당 해줌

3. A와 B를 구성하는 원소들의 값을 입력 받기 위해 이중 포문으로 입력을 받음. 행렬 사이에 \*를 제거하기 위해 cleanBuf 함수를 호출

4. 명령어의 개수인 m을 입력 받고 m번만큼 for문을 반복함

5. for문 내부에서 명령어를 cmd 배열에 저장하고 각 명령(ADD, TRANSPOSE, MULTIPLY)의 첫 글자는 A, T, M으로 다름으로 cmd[0]을 기준으로 switch문을 구성

6-1. cmd[0]이 A인 경우 인자를 체크하고 인자와 함께 ADD 함수를 호출

6-2. cmd[0]이 M인경우 MULTIPLY(A, B)와 MULTIPLY(B, A)가 다름으로 인자를 체크하고 그 인자를 기준으로 MUL함수를 호출

6-3. cmd[0]이 T인경우 인자를 체크하고 그 인자와 함께 TRN 함수를 호출

7. a와 b를 free 해주고 rEad와 wRite를 fclose로 닫아줌.

2. cleanBuf 함수

1. 파일 입출력을 위한 입력변수 rEad를 인자로 받음

2. 입력 받기 위한 char형 변수 buf를 선언

3. buf에 한 글자씩 fscanf로 입력을 받다가 buf==’\*’인 순간 멈추고 0을 리턴 함. (\*로 input section이 구분되어 있기 때문에 \*가 a, b등의 행렬이나 cmd에 잘못 입력되는 것을 막기 위해서)

3. ADD 함수

1. 파일 입출력을 위한 출력변수 wRite, 입력받은 행렬을 저장한 int 더블포인터형 변수 a, b, 행렬의 길이인 n을 인자로 받음

2. 행렬들의 합을 저장할 New라는 int형 더블포인터 변수를 선언하고 n x n크기를 동적할당, non-zero 값의 수를 세어 줄 t를 선언하고 0으로 초기화

3. 인덱스 문자인 i, j를 선언하고 이로 이중 for문을 돌려서 행렬의 덧셈의 정의대로 = 가 되게 함. 이때 가 0이 아닐 경우 t를 증가시킴

4-1. t가 보다 클 경우 New 행렬을 인자로 printMatrix 함수를 호출

4-2. t가 보다 작을 경우 sparse 함수를 호출

4. MUL 함수

1. 파일 입출력을 위한 출력변수 wRite, 입력받은 행렬을 저장한 int 더블포인터형 변수 a, b, 행렬의 길이인 n을 인자로 받음

2. 행렬들의 곱을 저장할 New라는 int형 더블포인터 변수를 선언하고 n x n크기를 동적할당, non-zero 값의 수를 세어 줄 t를 선언하고 0으로 초기화

3. 행렬의 곱셈의 정의에 따라 = 임으로 삼중 for문을 이용하여 A와 B의 곱행렬인 New 행렬을 계산, 이때 가 0이 아닐 경우 t를 증가시킴

4-1. t가 보다 클 경우 New 행렬을 인자로 printMatrix 함수를 호출

4-2. t가 보다 작을 경우 sparse 함수를 호출

5. TRN 함수

1. 파일 입출력을 위한 출력변수 wRite, 입력받은 행렬을 저장한 int 더블포인터형 변수 a, 행렬의 길이인 n을 인자로 받음

2. 행렬들의 곱을 저장할 New라는 int형 더블포인터 변수를 선언하고 n x n크기를 동적할당, non-zero 값의 수를 세어 줄 t를 선언하고 0으로 초기화

3. Transpose 의 정의에 따라 를 이중 for문으로 구현 후 가 0이 아닐 경우 t를 증가시킴

4-1. t가 보다 클 경우 New 행렬을 인자로 printMatrix 함수를 호출

4-2. t가 보다 작을 경우 sparse 함수를 호출

6. printMatrix 함수

1. 파일 입출력을 위한 출력변수 wRite, 입력받은 행렬을 저장한 int 더블포인터형 변수 a, 행렬의 길이인 n을 인자로 받음

2. 이중 for문을 이용하여 A의 모든 인자를 fscanf로 출력함’

7. Sparse 함수

1. 파일 입출력을 위한 출력변수 wRite, 입력받은 행렬을 저장한 int 더블포인터형 변수 a, 행렬의 non-zero 값의 개수인 t를 인자로 받음

2. 주어진 행렬의 Sparse Matrix를 저장할 int형 더블포인터 변수 sp를 선언하고 3 x t크기를 동적할당

3. 이중 for문으로 주어진 행렬 A의 모든 원소를 검사하면서 0이 아닌 값이 나올 때마다 그 값의 행 좌표를 sp[0]에, 열 좌표를 sp[1]에, 그 값을 sp[2]에 저장함

4. 이중 for문으로 sp의 모든 값을 출력함

performance analysis

n x n의 행렬이 입력으로 주어졌을 때 최선의 경우는 MULTYPLY 명령어를 받지 않는 경우이다. 이때 ADD와 TRN 함수는 i, j 두 변수를 n번 반복하는 for문을 중첩해서 사용함으로 시간 복잡도는 을 가진다.

반면 최악의 경우는 MULTIPLY 함수를 포함해서 입력을 받는 경우인데 이때 MULTIPLY 함수는 i, j, k 세 변수를 0부터 n-1까지 반복해서 연산을 하기 때문에 이 알고리즘은 최악의 경우 의 시간 복잡도를 가진다.