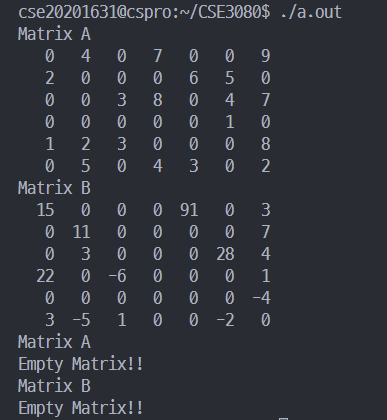
**오픈랩 과제 #4 : Matrix with linked list representation**

20201631 장유빈

**과제 실행 결과**



**알고리즘 설명**

본 알고리즘은 희소행렬을 효율적으로 메모리에 저장하고, 또 사용하기 위한 알고리즘이다. 희소행렬은 행렬의 값이 대부분 0이므로 0이 아닌 숫자가 있는 곳만 표시해주는 것이 효율적이다. 이때 사용할 자료구조는 linked list이다.

노드는 크게 두 종류로 구성되어 있다. 먼저 각 row, 또는 column을 대표하는 노드인 head node가 있는데, 이 노드는 각 row 또는 column 안의 0이 아닌 행렬 entry에 접근할 수 있고, 다음 row 또는 column을 가리키는 포인터를 포함하고 있다. 다른 노드는 entry node인데, entry에 대한 정보인 row, column, value와 함께 오른쪽 노드, 아래의 노드에 대한 포인터를 포함하고 있다. 이와 함께 tag라는 값이 하나 더 존재하는데, 이 값은 이 노드가 entry node인지 head node인지 구분해준다. 여기서의 head node는 앞에서 언급한 head node와 달리, head node의 head node이다. 즉, head node들에 접근하는 노드인 것이다. 즉, head node는 entry node로 구성된 노드들 또는 다음 head node를 가리킨다. 그리고 head node의 head node는 entry node의 형태를 띄고 있다. 또한 모든 list는 circular 형태를 띄고 있어서, 리스트의 마지막 노드에서 이어지는 다음 노드는 리스트의 첫 노드이다. row에 대한 head node에서 생각해본다면, 같은 row에 있는 모든 0이 아닌 entry를 다 리스트에 기록하고, 가장 column 값이 높은 entry의 포인터가 가리키는 노드는 head node가 되는 것이다.

본 알고리즘은 3개의 주요 함수로 이루어진다. 파일로부터(또는 키보드의 입력으로부터) 행렬을 읽어오는 함수인 mread(), 희소행렬을 2차원의 행렬형태로 다시 출력하는 mwrite(), 행렬이 있는 리스트를 삭제하는 merase()함수가 있다. 자세한 설명은 아래 코드의 주석으로 대신한다.

**코드**

설명은 주석을 통해 확인할 수 있다.

