Project 1: Using linked lists

Background

Sparse matrix is a matrix in which most elements are zero. A list data structure is used to represent the non-zero values of the matrix in a node. A node in the linked list contains the row index, column index and the non-zero value at their intersection. For instance, in a sparse matrix as shown below, the list will contain only 7 nodes representing the 7 non-zero elements of the sparse matrix.

0	0	0	0	0	0	0	0	0	0
0	**4**	0	0	0	0	0	**5**	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	**5**	0	0	0	0	**9**
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	**1**	0
0	0	**3**	0	0	0	0	0	0	0
0	0	0	0	0	0	**7**	0	0	0

Storing only the non-zero elements in the list helps in managing storage requirements efficiently. Imagine 100,000 rows and 1,000 columns with very few cells holding non-zero values. Instead of occupying 100,000 * 1,000 * 4 bytes is a lot of memory when most values are 0.

Some good uses of a sparse matrix are:

1. User ratings of movies or products. Not all users will watch or buy all movies or products.
2. Representation of a graph using adjacency matrix where the nodes far exceed the number of edges.

Your problem definition

The Indian Premier League (IPL) is a cricketing event in India involving 10 teams of 12 players (including the 12th player), most players playing a total of 74 matches. The league matches themselves account for 70 of them.

You are given two sparse matrix as inputs, each matrix containing 120 rows (each row represents a player) and 74 columns (each column represents an IPL match). The data provided to you is at the end of an IPL season.

The 1st matrix contains either a 1 or 0 indicating if a player has scored a half century or not in a match. For instance, in IPL 2024 only 15 players scored half centuries in the entire IPL season. Indeed, a very good case for building a sparse matrix. The 2nd matrix contains either a 1 or a 0 indicating if a player has taken a wickets in a match. Both matrices have 120 * 74 cells. Extrapolate this figure to 17 IPL seasons held so far. That is approximately (120 * 17) * (74 * 17) cells, which is a pretty large sparse matrix.

You are required to do the following to solve the problem:

1. Read the two sets of data from the files shared with you.
2. Create two linked lists representing the two sparse matrices with only the non-zero values stored in the nodes of the linked list. You can use either the row major or column major form to create the linked lists. Choose the same form to create both the lists.
3. Create two linked lists that stores the transpose of the two matrices defined above.
4. Create another linked list of nodes that stores the player number, total number of half centuries and total number of wickets taken. A player who has not scored a half century or has not taken a wicket will not feature in this linked list. This is a linked list of only players who are called 'all-rounders'. All-rounders are those players who can bat and bowl.
5. Display the sparse matrix (original and transposed) in its original form using the linked list representation.
6. What is a matrix called when the zero elements are very few? In what situations are such matrices useful? Can the same linked list implementation help in managing memory efficiently? Write a short note explaining your views on the above.