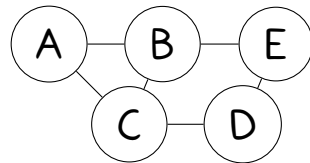


Efficient Representation of Social Networks

You're studying social networks and need to find an efficient way to store and process connection data. Let's explore different representations using this small social network:



Question 1: Create a 5x5 grid where both rows and columns represent individuals (A to E). Mark friendships with a 1, and non-friendships with a 0. What are the advantages and disadvantages of this representation?

Question 2: For each person, create a list of their friends' IDs. For example, A's list would be [B, C]. How does this representation compare to the grid in terms of efficiency?

Question 3: We can optimize this representation further. Create three lists:

1. A combined list, called indices, of all friends in order of A, B, C, D, E. For example, the first five elements of the combined list

are [B, C, A, C, E, ...], which is a concatenation of A's friends [B, C], and B's friends [A, C, E].

2. A list, called `pointers`, that shows where each person's friend list starts in the combined friend list. For example, if A has 2 friends and B has 3 friends and we combine the lists in order of A, B, C, D, E, B's list would start at index 2 in the combined list, and C's list would start at index 5.
3. A list, called `data`, of 1's with the same length as the combined friend list

Hint: Start the first list with 0, and for each subsequent entry, add the number of friends the previous person has.

Question 4: Using the representation you created in Question 3, how many friends does C have using the `pointres` list?

Question 5: List D's friends using the `pointers` and `indices` lists.

Question 6: How many total friendships are in the network using the `data` list?

Question 7: A makes a new friend with E. How would you modify your representation to reflect this change?