1.11 Graf quadrat

[CLRS 22.1-5 pàg 530]

Answer: The edge (u, w) exists in the square of a graph G if there exists a vertex v such that the edges (u, v) and (v, w) exist in G. To calculate this efficiently from an adjacency matrix, we notice that this condition is exactly what we get when we square the matrix. The cell $M^2[u, w] = \sum_v M[u, v] \cdot M[v, w]$ when we multiply two matrices. So, if we represent edges present in G with ones and all other entries as zeroes, we will get the square of the matrix with zeroes when edges aren't in the graph G^2 and positive integers representing the number of paths of length exactly two for edges that are in G^2. Using Strassen's algorithm or other more sophisticated matrix multiplication algorithms, we can compute this in $O(V^{2.376})$. Using adjacency lists, we need to loop over all edges in the graph G. For each edge (u, v), we will look at the adjacency list of v for all edges (v, w) and add the edge (u, w) to the adjacency lists for G^2. The maximum number of edges in the adjacency list for v is V, so the total running time is $O(VE)$. This assumes that we can add and resolve conflicts when inserting into the adjacency lists for G^2 in constant time. We can do this by having hash tables for each vertex instead of linked lists.