Problem numbers are from the second edition or the third of “Introduction to algorithms”. If unsure about which problem to solve, ask. Collaboration is permitted; looking for solutions from external sources (books, the web, etc.) is prohibited.

1. (a) Suppose that we are storing a set of \( n \) keys into a hash table of size \( m \). Show that if the keys are drawn from a universe \( U \) with \( |U| > mn \), then \( U \) has a subset of size \( n \) consisting of keys that all hash to the same slot, so that the worst-case searching time for hashing with chaining is \( \Omega(n) \).

(b) Suppose we wish to search a linked list of length \( n \), where each element contains a key \( k \) along with a hash value \( h(k) \). Each key is a long character string. How might we take advantage of the hash values when searching the list for an element with a given key?

(c) Given a graph \( G = (V, E) \), show that if \( v_1, \ldots, v_k \in V \) is the sequence of vertices of a shortest path between \( v_1 \) and \( v_k \), then \( v_1, \ldots, v_{k-1} \) is the sequence of vertices of a shortest path between \( v_1 \) and \( v_{k-1} \).

2. (a) 22.3-2

(b) 22.4-1

3. Describe an \( O(|V| + |E|) \) algorithm for the following problem: Given an undirected graph \( G = (V, E) \) as adjacency lists, determine whether we can paint each vertex red or blue so that adjacent vertices get different colors. If such a coloring exists, the algorithm outputs one such coloring. (Hint: Breadth-first search).