1 NPC

Do Exercise 34.5-1 (p.1100): the subgraph isomorphism problem.

2 NPC

Do Exercise 34.5-6 (p.1101). Here is an alternative description. In class, we will show the HAM-CYCLE problem is NP complete. Now, we consider a related problem: HAM-PATH. HAM-PATH problem is similar to HAM-CYCLE: it asks for a path of nodes for graph \( G \), such that the path visits each node exactly once. Note the difference is that in HAM-PATH, we do not need to return the starting node. Now show HAM-PATH is NP complete.

3 NPC

I have mentioned graph 3-coloring problem several times in class, but never got chance to prove the 3-coloring problem is NP complete (indeed it is, and one problem in Chapter 34 asks you to show it). But we are not going to do it here.

What I want you to do is to, assuming graph 3-coloring is NP complete, show graph 4-coloring is also NP complete. In 4-coloring problem, we want to color the nodes of graph with four colors and we still need to ensure no edge receives identical colors at its two ends. Again, we assume graph 3-coloring is known to be NP complete.

4 Reducing the LCS problem to the LIS problem

You can find reduction between two problems in \( \mathcal{P} \). Recall that we have fairly thoroughly studied the longest common subsequence (LCS) and the longest increasing subsequence (LIS) problem. Now the following reduces the LCS problem to the LIS problem.

Given two strings \( S_1 \) and \( S_2 \), I create a list \( L_x \) for each character \( x \) in the alphabet that appears in \( S_1 \), where \( L_x \) stores the list of positions that \( x \) appears in \( S_2 \) in the decreasing order. For example, let \( S_1 = abacx \) and \( S_2 = baabca \). Then, \( L_a = \{6, 3, 2\} \), \( L_b = \{4, 1\} \), \( L_c = \{5\} \) and \( L_x = \{\} \). Now, we create a list called \( \Pi(S_1, S_2) \), in which each character instance of \( x \) in \( S_1 \) is replaced by \( L_x \). That is, for each position \( i \) in \( S_1 \), insert \( L_{S_1[i]} \). In our example, \( \Pi(S_1, S_2) \) is \( \{6, 3, 2, 4, 1, 6, 3, 2, 5\} \).

Now, your task is to show that the LCS problem is polynomial-time reducible to the LIS problem. That is, you need to explain why a LCS of \( S_1 \) and \( S_2 \) corresponds to the LIS of \( \Pi(S_1, S_2) \).