CSC364 Summer 2002 – Homework 4

The following questions are assigned each week. None of these questions are for hand in, though you are encouraged to try them all.

Week 12

1. [CLRS, Problem 34-2, page 1018]
   Bonnie and Clyde have just robbed a bank. They have a bag of money and want to divide it up. For each of the following scenarios, either give a polynomial-time algorithm, or prove that the problem is **NP**-complete. The input in each case is a list of the n items in the bag, along with the value of each.

   (a) There are n coins, but only 2 different denominations: some coins are worth x dollars, and some are worth y dollars. They wish to divide the money exactly evenly.

   (b) There are n coins with an arbitrary number of different denominations, but each denomination is a nonnegative integer power of 2, i.e., the possible denominations are 1 dollar, 2 dollars, 4 dollars, etc. They wish to divide the money exactly evenly.

   (c) There are n checks, which are, in an amazing coincidence, made out to "Bonnie or Clyde." They wish to divide the checks so that they each get the exact same amount of money.

   (d) There are n checks as in part (c), but this time they are willing to accept a split in which the difference is no greater than 100 dollars.

2. We define the following three related problems.

5-CLIQUE
Instance: $\langle G \rangle$ G is a undirected graph.
Acceptance Condition: Accept iff G contains a 5-clique.

CLIQUE
Instance: $\langle G, k \rangle$, G is a undirected graph, k is an integer in binary.
Acceptance Condition: Accept iff G has a k-clique.

MAX-CLIQUE
Instance: $\langle G \rangle$, G is an undirected graph.
Goal: Return the size of the largest clique of G.

(a) Give a polynomial time algorithm that solves 5-CLIQUE.

(b) Prove MAX-CLIQUE $\rightarrow_p$ CLIQUE (recall that $\rightarrow_p$ is a polynomial time Turing reduction)

This proves we can solve MAX-CLIQUE with a polynomial number of calls to CLIQUE, and that if CLIQUE is in **P**, then MAX-CLIQUE is in **FP**. (Note that CLIQUE is proven **NP**-complete in the textbook.)

(c) Prove CLIQUE $\rightarrow_p$ MAX-CLIQUE.

This shows that MAX-CLIQUE is likely not in **FP** since CLIQUE is **NP**-complete.

(d) Consider the following algorithm for solving MAX-CLIQUE:

```
MAX-CLIQUE(G):
for i ← 1 to n do          // n is the number of vertices in G
  if not i-CLIQUE(G) then
    return i-1
return n
```

We proved in part (a) that i-CLIQUE is in **P**, and we only make a polynomial number of calls to k-CLIQUE. Why does this not prove MAX-CLIQUE is in **FP**?

Week 13

3. We have seen in class that GRAPH 3-COLOURABILITY is **NP**-complete.

   (a) Prove that GRAPH 2-COLOURABILITY is in **P**.
   Hint: Try a greedy algorithm.

   (b) Define GRAPH k-COLOURABILITY as:
   Instance: $\langle G, k\rangle$, G is an undirected graph, k an integer in binary.
   Acceptance Condition: Accept iff G if k-colourable, i.e., colours 1 through k can be assigned to the vertices of G such that no two adjacent vertices get the same colour.

   Prove that GRAPH k-COLOURABILITY is **NP**-complete.
   Hint: GRAPH 3-COLOURABILITY is **NP**-complete.

   (c) Let us restrict the input graphs to be trees. Prove that this restricted version of GRAPH k-COLOURABILITY is in **P**.

   (d) Explain, in no more than three sentences, how it is possible that GRAPH k-COLOURABILITY could be **NP**-complete for general graphs, yet be polynomial for certain restricted classes of graphs.

4. Define the class co-**NP** $= \{\overline{L} \mid L \in \mathbf{NP}\}$.
   In other words, if a problem X belongs to **NP**, its complement $\overline{X}$ (all its "no" instances) belongs to co-**NP**.

   For example,
   COMPOSITE $= \{m > 1 \mid m$ is not prime$\}$ is a problem in **NP**, so
   $\overline{\text{COMPOSITE}}$ = PRIME $= \{m > 1 \mid m$ is prime$\}$ is in co-**NP**.

   We can think of decision problems in co-**NP** as those which we can verify "no" instances quickly (there exist a polynomial sized certificate which can be verified in polynomial time for the "no" instances).

   (a) HC (Hamiltonian cycle) is a problem we know is in **NP**. It is not known whether $\overline{\text{HC}}$ is in **NP** or not, and most researchers suspect it is not. What kind of evidence could prove that a graph does *not* have a Hamiltonian cycle? Could it be succinct (polynomial size)?

   (b) Assuming that $\mathbf{NP} \neq$ co-$\mathbf{NP}$, prove that no **NP**-complete problem can belong to co-**NP**.

   (c) Define the class co-**P** analogously. Prove that $\mathbf{P} =$ co-$\mathbf{P}$.