Design and Analysis of Algorithms

Exercises 6/8

1. Colouring problem can be solved by non-deterministic algorithm as shown below. Convert it to a deterministic algorithm using decision tree and suitable search technique.

```
Colouring(G, k):
// Step 1: Oracle guess of the solution
FOR i:=1 TO n DO
   Colour[i] := CHOOSE(1, k);

// Step 2: Verify the solution
FOR i:=1 TO n DO
   FOR j:=1 TO n DO
     IF Edge(i, j) AND Colour[i]=Colour[j] THEN FAILURE;
SUCCESS;
```

2. Give a **non-deterministic** algorithm to solve *Sudoku*. What is the size of input? Derive time complexity of your algorithm? Does it prove that Sudoku is NP hard?

3. *Traveling salesman problem* (TSP) is defined as an *optimization problem*, to find the shortest tour that visits all the nodes exactly once. Two closely related decision problems can be defined as:

a. TSP-1: does there exist a traveling salesman tour that has length smaller than *k*?
b. TSP-2: does there exists traveling salesman tour that has length equal to exactly *k*?

Suppose that we have algorithms solving these two problems: A_1 for TSP-1 and A_2 for TSP-2. Can you use these algorithms to solve the original optimization problems? If yes, show how.

4. Consider the following problems: *SAT* (satisfiability problem), CNF_3 (satisfiability of conjective normal form with 3 literals), *COL* (coloring problem), *EC* (exact cover problem), *KP* (knapsack problem), and *TSP* (traveling salesman problem). The following chain of reductions is known: $SAT \leq_P CNF_3 \leq_P COL \leq_P EC \leq_P KP \leq_P TSP$. What can you say about the complexities of *SAT* and *TSP* relative to each other? Specifically, what can you say about the claims $SAT \leq_P TSP$, and $TSP \leq_P SAT$.

5. We have three problems L_1, L_2 and L_3 which all can be solved in polynomial time. The first one requires non-deterministic algorithm, but we don't have access to it right now. For the other two we have A_2 that is non-deterministic and A_3 that is deterministic. Problem L_x is proven to be *NP-complete*. Assume that the following reductions exist: $L_1 \leq_P L_x$, $L_2 \leq_P L_x$ and $L_x \leq_P L_3$. Can we solve problem L_1 with a 10 000 GHz computer? What else can you conclude from this information?

6. Consider an instance of the *knapsack problem*, where the size of the knapsack is 11 and the numbers to be put into the knapsack are 1, 2, 4, 8 and 16. Convert the input into an input to the travelling salesman problem. Mark the path that corresponds to the solution of the knapsack problem onto the graph.