CSCE 340 STUDY GUIDE 4

DR. GODFREY MUGANDA

Do not look up solutions for the study guide on the Internet: that defeats the purpose of you trying to figure it out yourself. Come for help if you need it!

1. Backtracking Algorithms

1. Assume that you have a directed graph specified by an adjacency list. You are given two vertices A and B.

Write an algorithm that prints *all* paths from the source vertex A to the destination vertex B. Consider only paths that go through each vertex at most once.

If you want to write a program implementing your algorithm, compare this problem to the *Firetruck* problem at the UVA online judge programming competition site.

2. Devise a backtracking algorithm for solving the 0-1 Knapsack problem (without repetition) on page 167 of the class textbook.

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. A subset $C \subset V$ is called a *vertex cover* of G if every edge of G has at least one of its end points in C.

3. Assume that a graph is specified by an adjacency matrix M, and that a set of vertices C is represented by a Java linked list. Write a Java method that returns **true** if C is a vertex cover and **false** otherwise.

4. Given an integer k where $1 \leq k \leq |V|$, devise a backtracking algorithm for computing a vertex cover of size at most k is one exists. Follow the backtracking framework we have been using in this course by writing a method

```
      boolean extend(C, k, s,  M)
```

where C is a linked list representing a *partial solution* at a stage s and M is the adjacency matrix.

2. Dynamic Programming

Do problems 6.1, 6.2, 6.3, 6.6, and 6.25 in the class textbook.

Do not feel discouraged if you can't figure all of them out. Just try to solve as many of them as you can. As you study for the quiz, the most important thing is to determine the recurrence and the boundary conditions.

Even if you cannot figure out a problem, you will be in a better position to understand the solution when you come for help.