Missouri University of Science & Technology — Department of Computer Science
Spring 2022 — **CS 2500: Algorithms**

Sample Questions for Exam 2

Name: *Sid Nadendla* — **Email:** *nadendla@mst.edu*

This handout presents a few review questions for Midterm 2 on April 28, 2022. Midterm 2 will be 1.5-hour long in-class exam with about 2-3 questions.

Section 1: Dynamic Programming

1. Consider a 0-1 Knapsack problem with n indivisible items, where v_i and w_i are the value and weight of the i^{th} item respectively. If the size of the Knapsack is W, show that the Bellman equation is

$$V[i,j] = \min_{x_i \in \{0,1\}} v_i x_i + V[i-1, j - x_i w_i], \text{ for all } i = 1, \cdots, n \text{ and } j = 0, 1, \cdots, W,$$

   and write the pseudocode for the dynamic programming solution to 0-1 Knapsack problem.

2. Consider a matrix chain multiplication $A_1 \cdot A_2 \cdots A_n$, where A_i is a $p_{i-1} \times p_i$ matrix. IF the cost of computing the product of $k \times \ell$ and $\ell \times m$ matrices is $k\ell m$, show that the Bellman equation is

$$m[i,j] = \begin{cases} m[i,k] + m[k+1,j] + p_{i-1}p_k p_j, & \text{if } i > j \\ 0, & \text{otherwise.} \end{cases}$$

   Using the above Bellman equation, write a pseudocode for the dynamic programming algorithm for matrix-chain multiplication problem.

Section 2: Greedy Algorithms

1. When can a greedy algorithm produce an optimal solution?

2. Demonstrate on the following example, how to find an optimal solution to the Fractional Knapsack problem using a greedy algorithm?

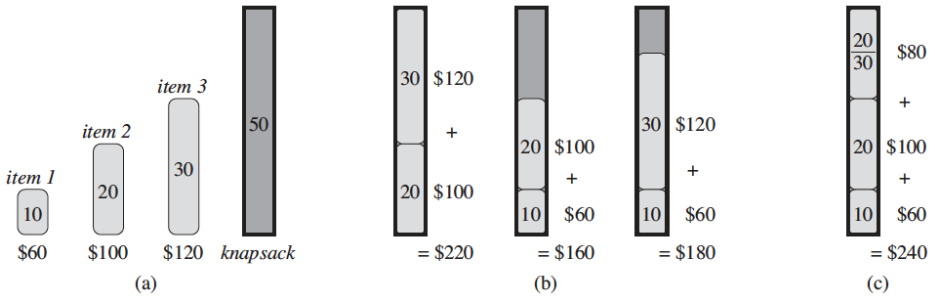

Figure 1: Fractional Knapsack

Section 3: Graph Search/Traversal

1. Demonstrate Breadth-first search (BFS) on a graph, such as the one shown in Figure 2.
2. Demonstrate Depth-first search (DFS) on a graph, such as the one shown in Figure 2.
3. What is the main difference between the implementation of BFS and DFS algorithms?

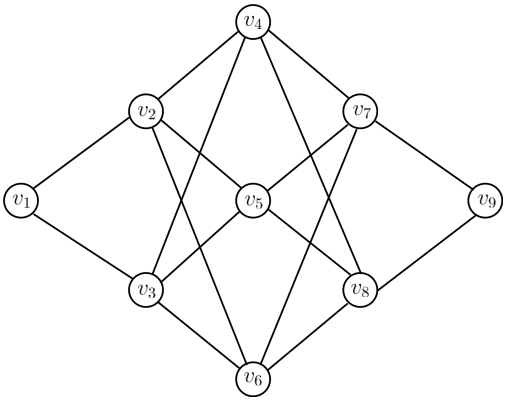

Figure 2: Graph Search

Section 4: Minimum Spanning Trees

1. Demonstrate Kruskal's algorithm on the graph, such as the one shown in Figure 3.
2. Demonstrate Prim's algorithm on the graph, such as the one shown in Figure 3.
3. Explain why dynamic programming is not a good approach to find minimum spanning trees?
4. State the primary difference between Kruskal's and Prim's algorithms?

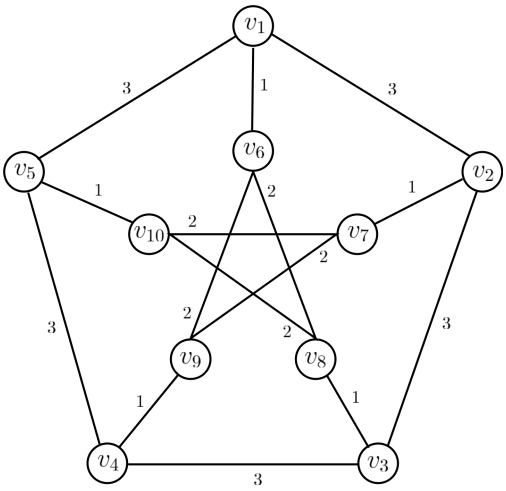

Figure 2: Minimum Spanning Trees