Final Review

1. Demonstrate that the Graph-Coloring Optimization problem is in NP.

2. Draw the entire pruned state space tree that is generated by the backtracking algorithm for the Sum of Subsets problem on the following instance. (15 points)

- W = 15
- $w_1 = 3$ $w_2 = 5$ $w_3 = 6$ $w_4 = 7$

Final Review Answers

1. Demonstrate that the Graph-Coloring Optimization problem is in NP.

Decision Problem: Given a graph G, can we color each node such that we use *m* or less colors and no adjacent node is the same color?

Verification Algorithm: Inputs: An adjacency matrix W, an array of color choices `color` where `color[i]` is the color chosen for the i^{th} vertex, and *m*, the max number of colors we can use.

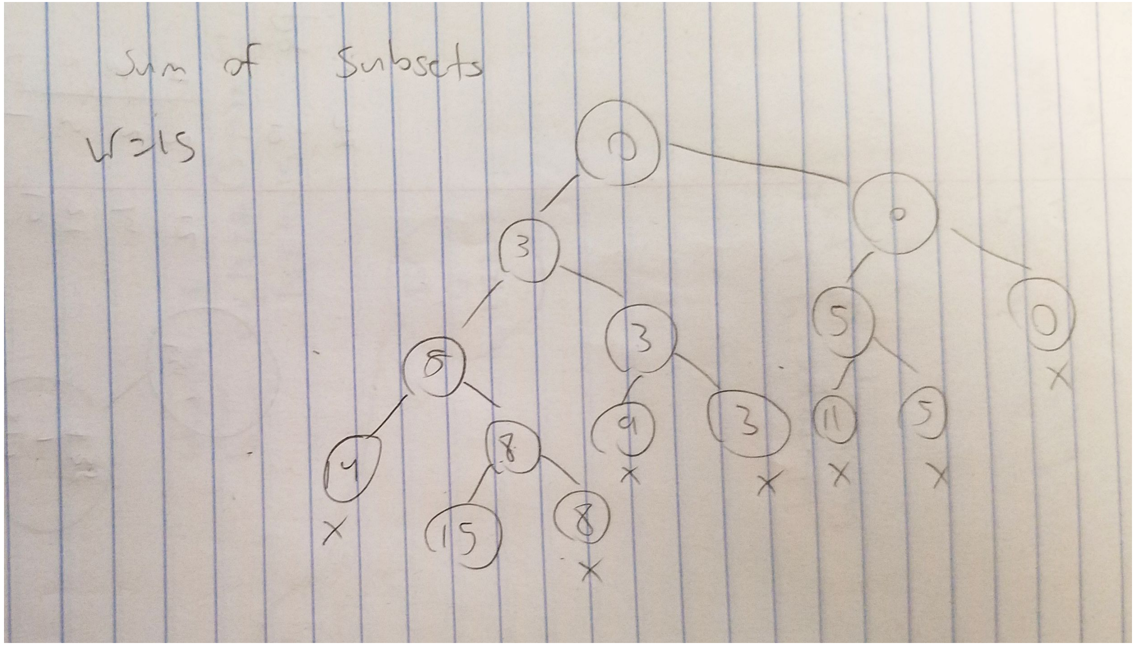
1. Verify that the guess is of the correct format.
2. Verify that every vertex is assigned a color.
3. Iterate through `color`. For each vertex, check W for its adjacent vertices. If any adjacent vertices are the same color return false. Keep a count of how many colors were used. If numColors > m, return false. Otherwise, return true.

This procedure runs in $O(n^2)$ time due to the nested loop (for each vertex, check if every other vertex is adjacent)

Final Review Answers

Draw the entire pruned state space tree that is generated by the backtracking algorithm for the Sum of Subsets problem on the following instance. (15 points)

- W = 15
- $w_1 = 3$ $w_2 = 5$ $w_3 = 6$ $w_4 = 7$



Final Review

3. Given the following cost adjacency matrix, use Floyd's Algorithm to compute $D^{(4)}$, thus finding the shortest distance from every vertex to every other vertex.

	1	2	3	4
1	0	8	∞	1
2	∞	0	1	∞
3	4	∞	0	∞
4	∞	2	9	0

Final Review Answer

$D^{(0)}$:

	1	2	3	4
1	0	8	∞	1
2	∞	0	1	∞
3	4	∞	0	∞
4	∞	2	9	0

$D^{(1)}$:

	1	2	3	4
1	0	8	∞	1
2	∞	0	1	∞
3	4	12	0	5
4	∞	2	9	0

$D^{(2)}$:

	1	2	3	4
1	0	8	9	1
2	∞	0	1	∞
3	4	12	0	5
4	∞	2	3	0

$D^{(3)}$:

	1	2	3	4
1	0	8	9	1
2	5	0	1	6
3	4	12	0	5
4	7	2	3	0

Final Review Answer

$D^{(3)}$:

	1	2	3	4
1	0	8	9	1
2	5	0	1	6
3	4	12	0	5
4	7	2	3	0

$D^{(4)}$:

	1	2	3	4
1	0	3	4	1
2	5	0	1	6
3	4	7	0	5
4	7	2	3	0

Final Review

4. Consider the following instance of the 0-1 knapsack problem. Solve it with the branch-and-bound solution discussed in class. Show the state space tree and the priority queue after each step.

W = 5

	w_i	p_i
1	2	12
2	1	10
3	3	21
4	2	14

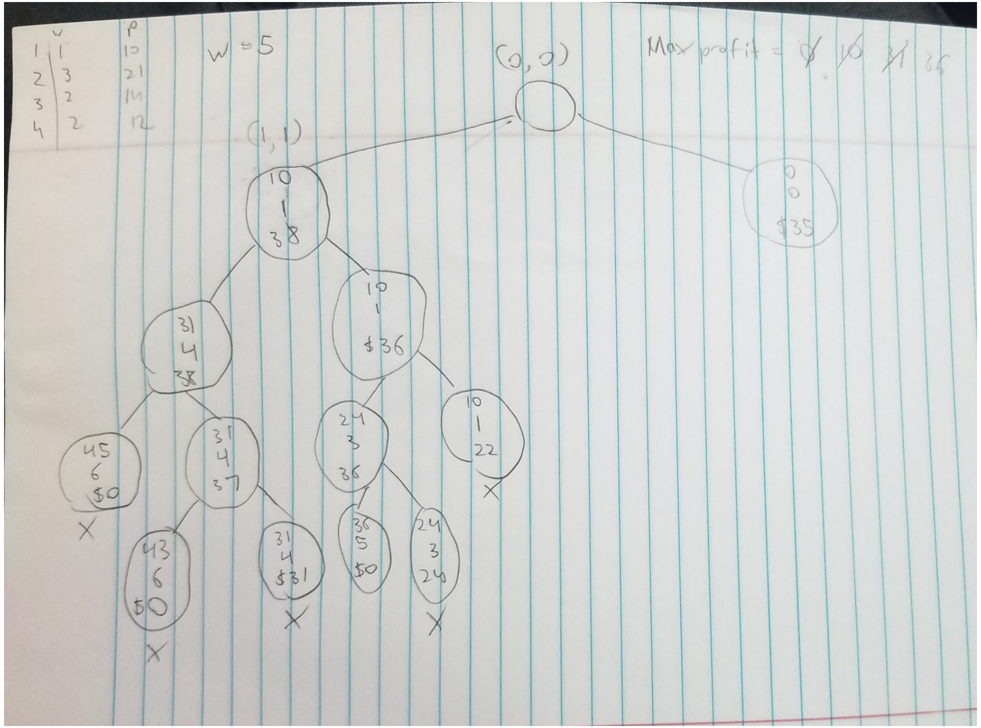
Final Review Answer

First step is to sort in order of ratio, profit / weight. Note that I changed the item numbers after sorting for simplicity (you don't have to do this)

	w_i	p_i
1	2	12
2	1	10
3	3	21
4	2	14

	w_i	p_i
1	1	10
2	3	21
3	2	14
4	2	12

Final Review Answer



I apologize for my handwriting!

Final Review

5. Draw the recursion tree formed by QuickSort on the following array. You may assume that the first element is selected as the pivot:

[5, 2, 8, 7, 6, 1, 2, 3, 9, 10, 5]