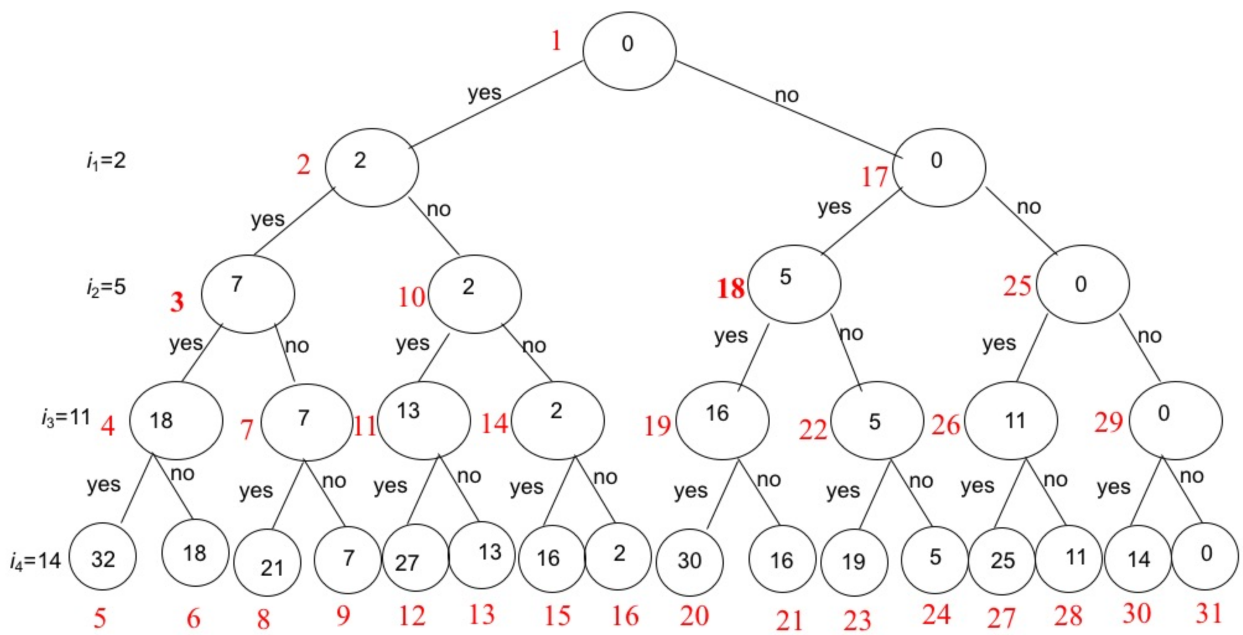
Design and Analysis of Algorithms
CS575, Fall 2017

Answers to Theory Assignment 4

1. [30%] A set of inters A = {2, 5, 11, 14} is given. For the given set A, find **every subset** that sums to S = 16.
    a. [15%] Solve the problem using the depth first method. Draw a state space tree and clearly show every step. Also, number the nodes in the sequence of visiting them.
    b. [15%] Find the subsets via backtracking. Draw a (pruned) state space tree and clearly show every step. Number the nodes in the sequence of visiting them too.

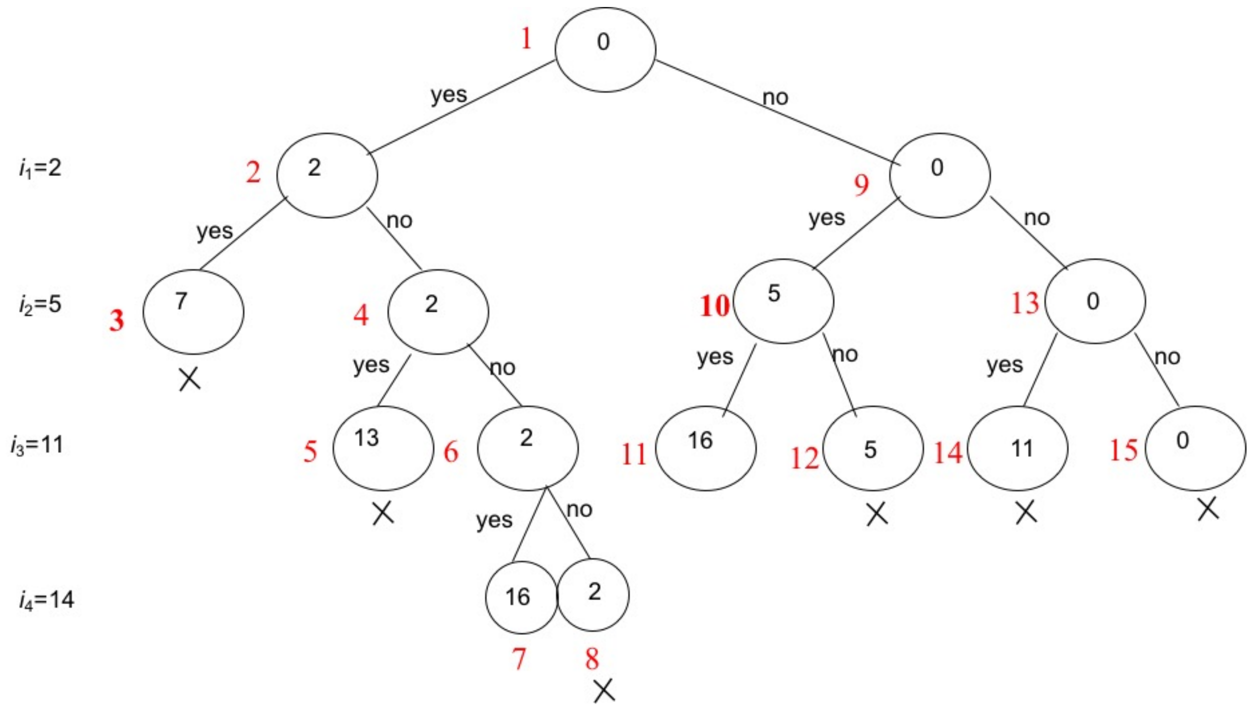
Answer:

a. There are two solutions: {2, 14} and {5, 11}.



Only the leaf nodes 15 and 21 satisfy that the sum of subset is equal to 16.

b. The weights are already sorted. Otherwise, you need to sort the weights (range 1 through n=4).



You need to show it for each node.
e.g. Node 4: weight = 2, weightSoFar =2, totalPossibleLeft=25. The Node is promising because weightSoFar + totalPossibleLeft = 27 $\geq$ 16 and weightSoFar + w[3] = 2+11=13 $\leq$ 16.
Node 5: weight = 13. The node is non-promising because weightSoFar + w[4] =27 > 16.
Node 8: weight = 2. The node is non-promising because weightSoFar + totalPossibleLeft =2+0=2 < 16.
Node 15: weight = 0. The node is non-promising because weightSoFar + totalPossibleLeft =14 < 16.

2. [10%] Consider the following revised sum of subsets problem: **Given *n* positive integers w_1, …, w_n and two positive integers S_1 and S_2, find *all* subsets of w_1, …, w_n that add to a number between S_1 and S_2 (including S_1 and S_2).** Suppose we follow the same general backtracking method for solving the original sum of the subsets problem (see the slides of the Backtracking chapter) and use the same notations for weightSoFar and totalPossibleLeft. Define the condition(s) for a node in the state space tree to be promising.

Answer: A node is promising if *both* of the following are true:

- *weightSoFar + totalPossibleLeft* $\geq S_1$
- *weightSoFar + wi+1 $\leq$ S2 or S1 $\leq$ weightSoFar $\leq$ S2*

3. [45%] When the capacity of the knapsack is 13, solve the following **0-1 knapsack** problem using (a) [15%] the backtracking algorithm; (b) [15%] the Breadth-First search with branch and bound pruning algorithm and (c) [15%] the Best-first search with branch and bound pruning algorithm in Chapter 13. All algorithms use the optimal fractional knapsack algorithm to compute the upper bound of the profit. Show the actions step by step.

i	p_i	w_i	p_i / w_i
1	\$20	2	\$10
2	\$30	5	\$6
3	\$35	7	\$5

4	$12	3	$4
5	$3	1	$3

Answer: See the end.

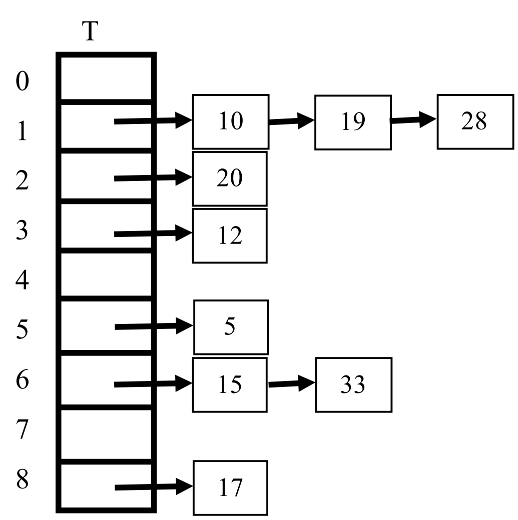
4. [15%] Assume that a hash table has 9 buckets where each bucket has only one slot. A simple hash function: home bucket = key % 9 (where % is a mod function) is used to compute the home bucket based on the key. You are supposed to insert the following keys to the hash table: 5, 28, 19, 15, 20, 33, 12, 17, 10 using the following overflow handling methods.
    a. [7%] Use the **linear probing** (linear open addressing) method to handle overflows, if any.
    b. [8%] Use the **sorted chaining** method to deal with overflows, if any.

Answer: $h(5) = 5$, $h(28) = 1$, $h(19) = 1$, $h(15) = 6$, $h(20) = 2$, $h(33) = 6$, $h(12) = 3$, $h(17) = 8$, $h(10) = 1$.

a.

0				4				8
10	28	19	20	12	5	15	33	17

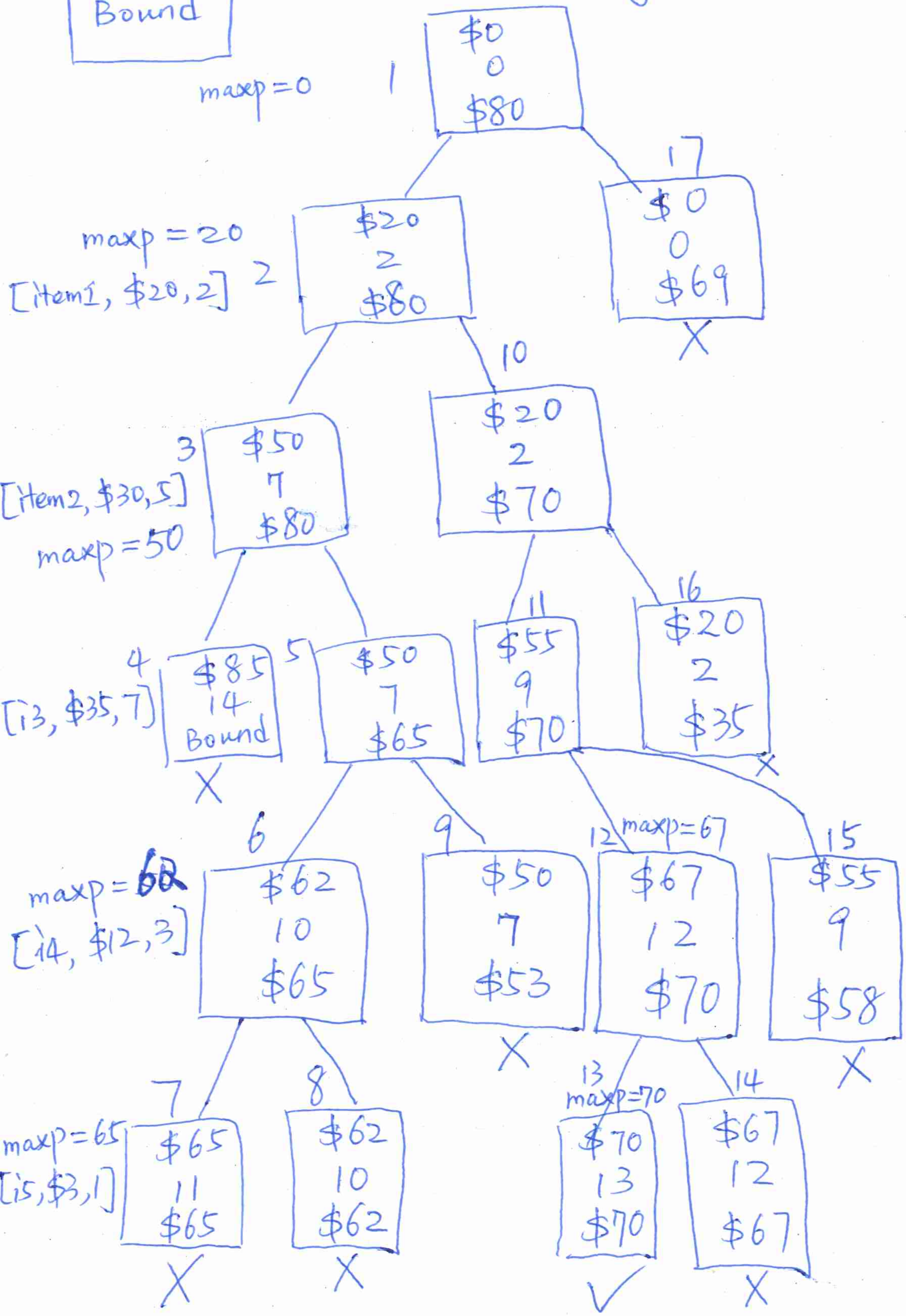
b.



Answer: 3(a)

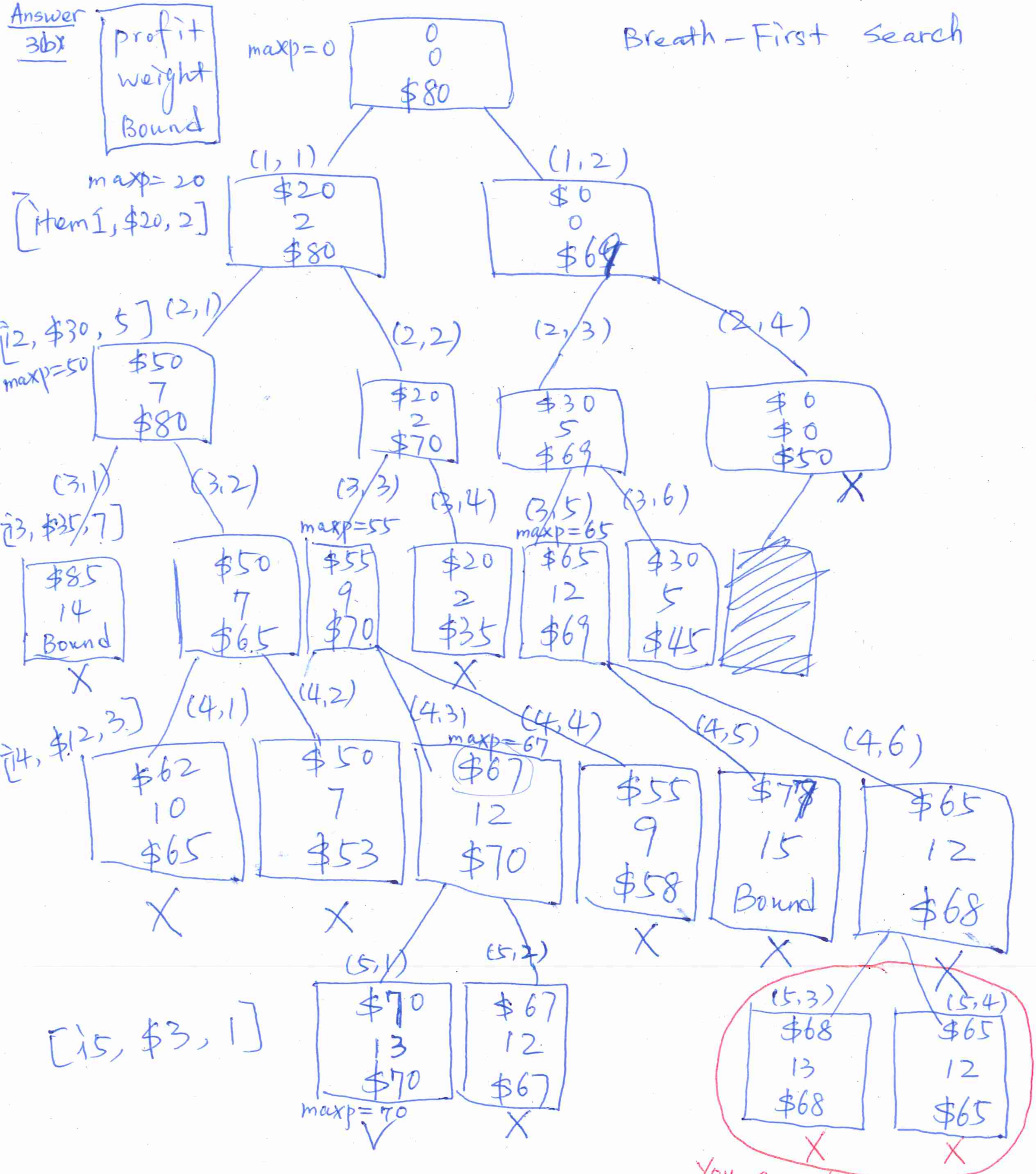
Profit
Weight
Bound

Depth-First Search
Backtracking Algorithm.



Answer 3(b)

Breath-First Search



You can have this or not depending on whether the algorithm checks again bound > maxprofit after putting in the queue.

Answer 3(c):

Best-First search