

## CS575 Homework 4

**Due at the Beginning of the Class on May 5**

**Submit a scanned pdf file through blackboard by 8:59am on May 5, and submit a hard copy at the beginning of the Class on May 5. (The soft and hard copies should be exactly the same.)**

**Your Name:** \_\_\_\_\_ **Section:** \_\_\_\_\_

I promise to follow the academic honesty requirements of the Binghamton University. I agree that I will fill out and sign an official form that I have cheated if I get caught cheating. I understand that this form will be stored by the university. Furthermore, I understand that the minimum penalty for cheating is getting a grade of 0 for this assignment.

**Sign:** \_\_\_\_\_

1. [30%] You are given a **0-1 knapsack** problem where the capacity of the knapsack  $W = 30$  and the set of items  $S = \{(i_1, 5, \$50), (i_2, 20, \$140), (i_3, 10, \$60)\}$  where each element in set  $S$  is a tuple of (item ID, weight, profit). Solve the given 0-1 knapsack problem using the dynamic programming method discussed in Chapter 12. Clearly show every step.
2. [40%] A set  $\{3, 4, 5, 6\}$  is given. For the set, find **every subset** that sums to  $S = 13$ .
  - a. [10%] 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. [30%] 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.
3. [30%] When the capacity of the knapsack is 16, solve the following **0-1 knapsack** problem using the backtracking algorithm discussed in class that uses the optimal fractional knapsack algorithm to compute the possible upper bound of the profit.

$i$	$p_i$	$w_i$	$p_i / w_i$
1	\$10	5	\$2
2	\$30	5	\$6
3	\$40	2	\$20
4	\$50	10	\$5

4. [20%] Assume that a hash table has 17 buckets where each bucket has only one slot. A simple hash function:  $\text{home bucket} = \text{key} \% 17$  (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: 6, 12, 34, 29, 28, 11, 23, 7, 0, 33, 30, 45 using the following overflow handling methods.
- (1) Use the linear probing (linear open addressing) method to handle overflows, if any.
  - (2) Use the sorted chaining method to deal with overflows, if any.