UWI

Design and Analysis of Algorithms

Tutorial 9 - Sample Solutions

Backtracking Technique

1. Explain the theory behind the backtracking technique.

   Backtracking eliminates some unnecessary cases from consideration and yields solutions in a reasonable time (mostly).

2. Solve the subset sum problem using the backtracking approach for the given set A = {1, 2 ,5, 6, 8}, with d as **9**.

   We proceed by constructing a binary tree where each level considers if we take a given item from set A or not.

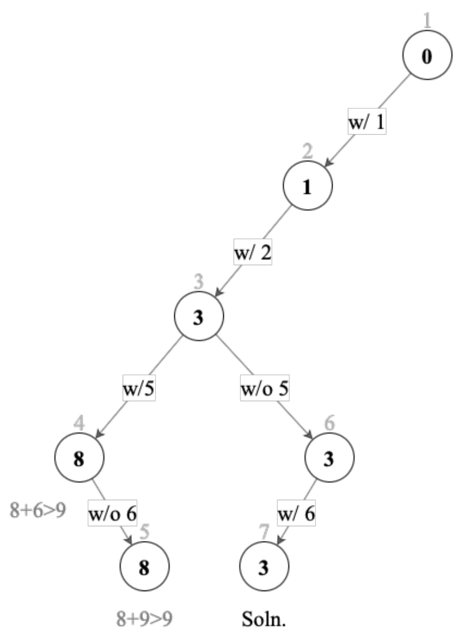

Figure 1: Sample solution to a small instance of the subset sum problem

From figure 1, we notice that the solution is a set containing {1,2,6}. We can stop here, unless we want to find all solutions. Notice the order of traversal, where we explore down to the left first then backtrack when we do not arrive at a solution, akin to in-order traversal.

3. Attempt to solve the 3-Queens problem using the backtracking technique. Is there a solution?

   We proceed by constructing a tree where each level considers each possible position of the next queen.

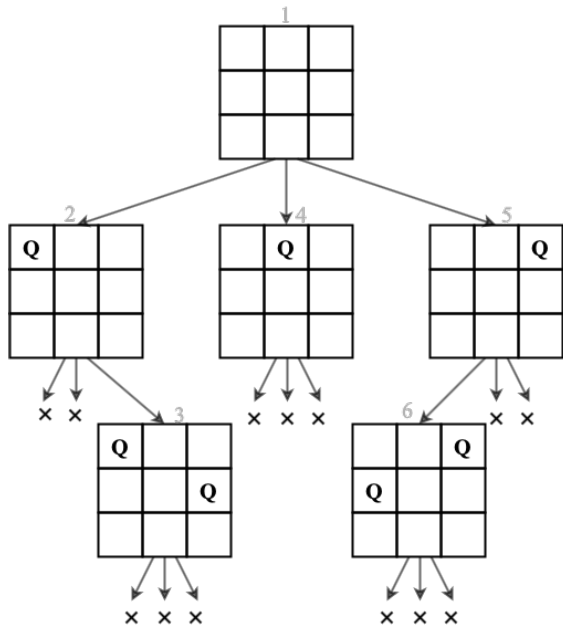

Figure 2: Sample solution to 3-queens problem

   From figure 2, we notice that there are no solutions to the 3-queens problem. No matter where we place the third queen, it will be in the diagonal or same column as another queen.

Branch-and-Bound Technique

1. Explain the theory behind the branch and bound technique.

   A further refinement of backtracking for optimization problems.

2. Solve the 0/1 knapsack problem using the branch-and-bound technique for the instance shown in table 1 with a knapsack capacity of **10**.

Item	Weight	Value
1	4	\$40
2	7	\$42
3	5	\$25
4	3	\$12

Table 1: A small instance of the 0/1 knapsack problem

   To solve the 0/1 knapsack problem using the branch-and-bound technique, we compute the value-to-weight ratios and select the items in decreasing order of these ratios. These calculations are shown in table 2.

Item	Weight	Value	$\frac{v_i}{w_i}$
1	4	\$40	10
2	7	\$42	6
3	5	\$25	5
4	3	\$12	4

Table 2: Value to weight ratios for the above instance of the knapsack problem

Next we calculate the upper bound of each node as we explore using the branch and bound technique using the following formula.

$$ub = v + (W - w)(\frac{v_{i+1}}{w_{i+1}}) \tag{1}$$

In the above formula, w is the total weight of the first i items.

$$ub(0) = 0 + (10 - 0) \times 10 = 100 \tag{2}$$

$$ub(1) = 40 + (10 - 4) \times 6 = 76 \tag{3}$$

$$ub(2) = 0 + (10 - 0) \times 6 = 60 \tag{4}$$

$$ub(4) = 40 + (10 - 4) \times 5 = 70 \tag{5}$$

$$ub(5) = 65 + (10 - 1) \times 4 = 69 \tag{6}$$

$$ub(6) = 40 + (10 - 4) \times 4 = 64 \tag{7}$$

$$\tag{8}$$

We end up with the following figure.

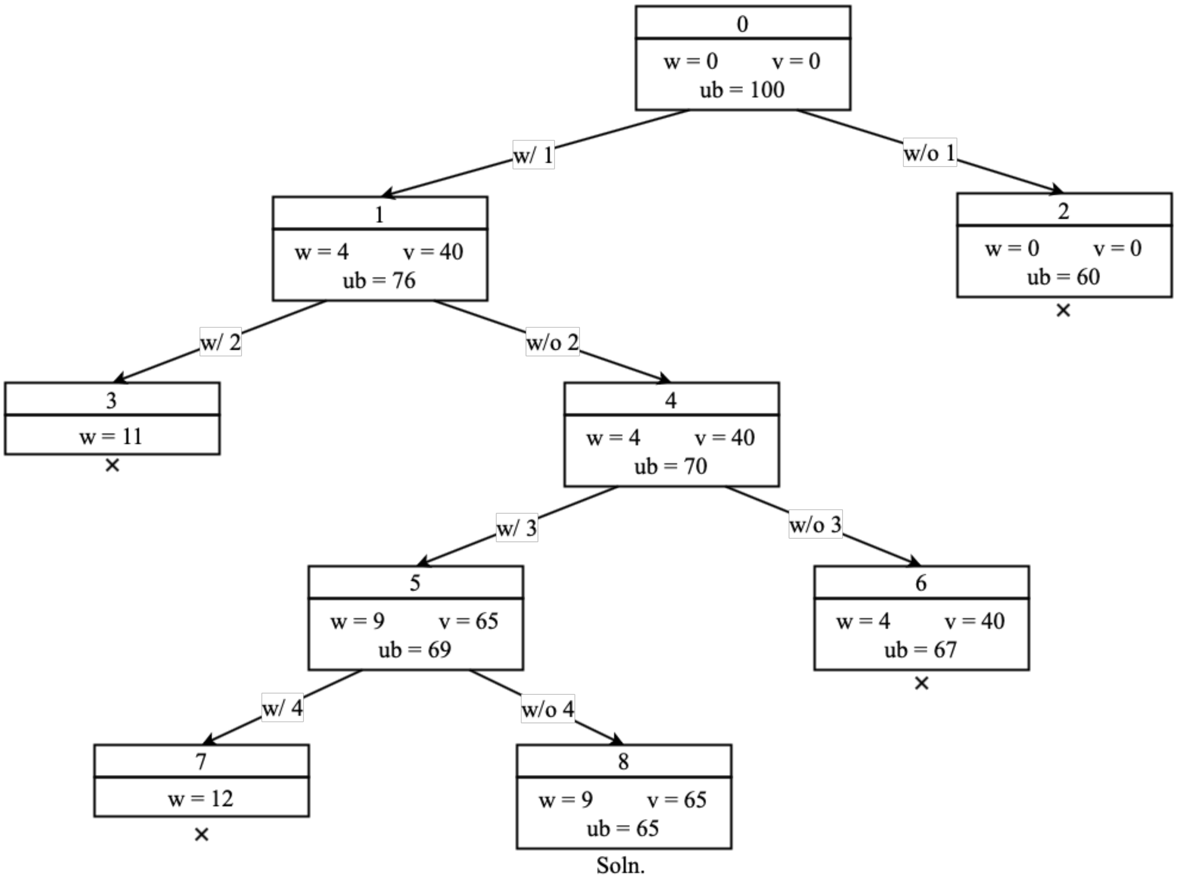

Figure 3: Sample solution to above instance of the knapsack problem

We observe that the solution is a set containing items 1 and 3.