

# Sum of subsets problem



## 5.4 The Sum-of-Subsets Problem

- given  $n$  positive integers (weights)  $w_1, w_2, \dots, w_n$
- given a positive integer  $W$
- finding all subsets of  $n$  integers that sum to  $W$ 
  - e.g.,  $w_i + w_j + \dots + w_k = W$

# Sum of subsets problem



## 5.4 The Sum-of-Subsets Problem

- create a state space tree
- each ***left*** edge denotes we ***include***  $w_i$  (weight  $w_i$ )
- each ***right*** edge denotes we ***exclude***  $w_i$  (weight 0)
- any path from root to a leaf forms a subset

# State space tree

## 5.4 The Sum-of-Subsets Problem

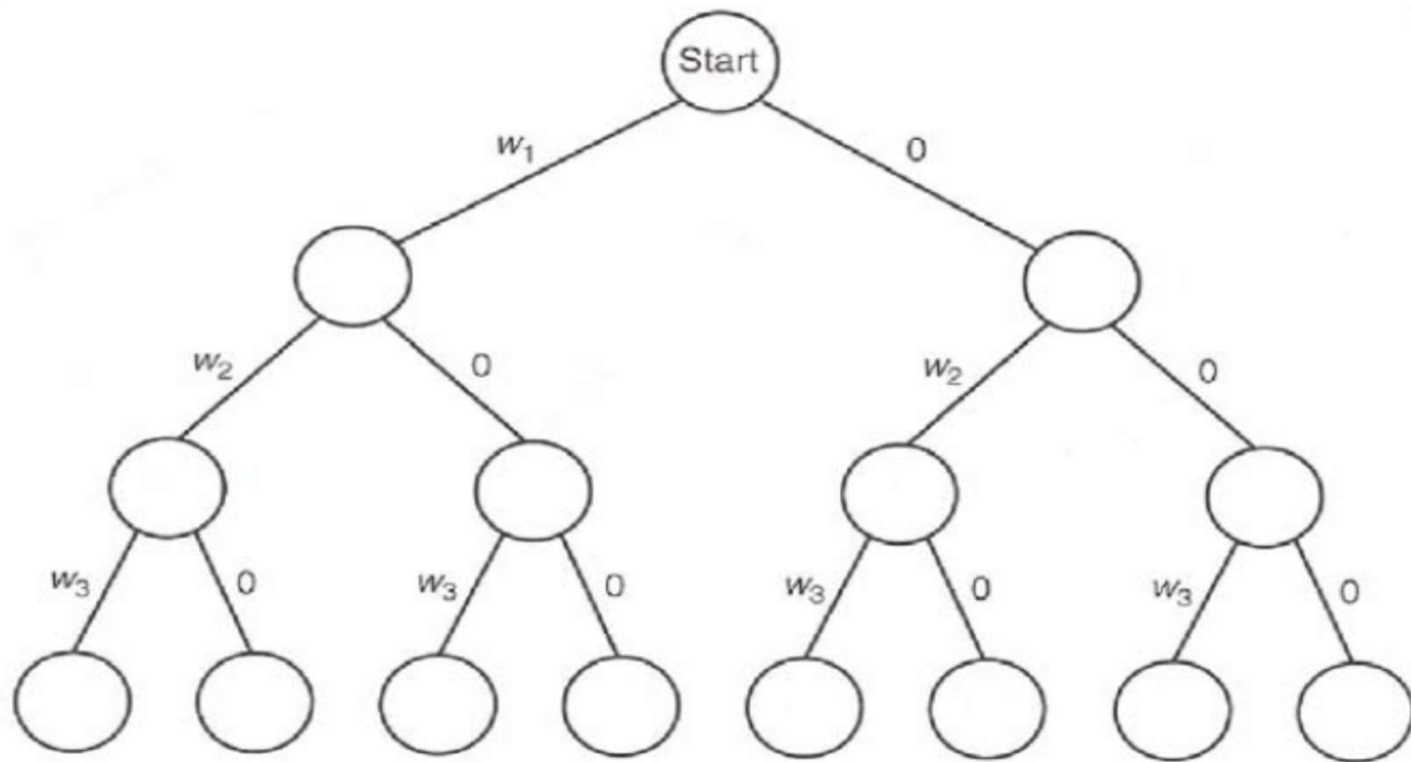
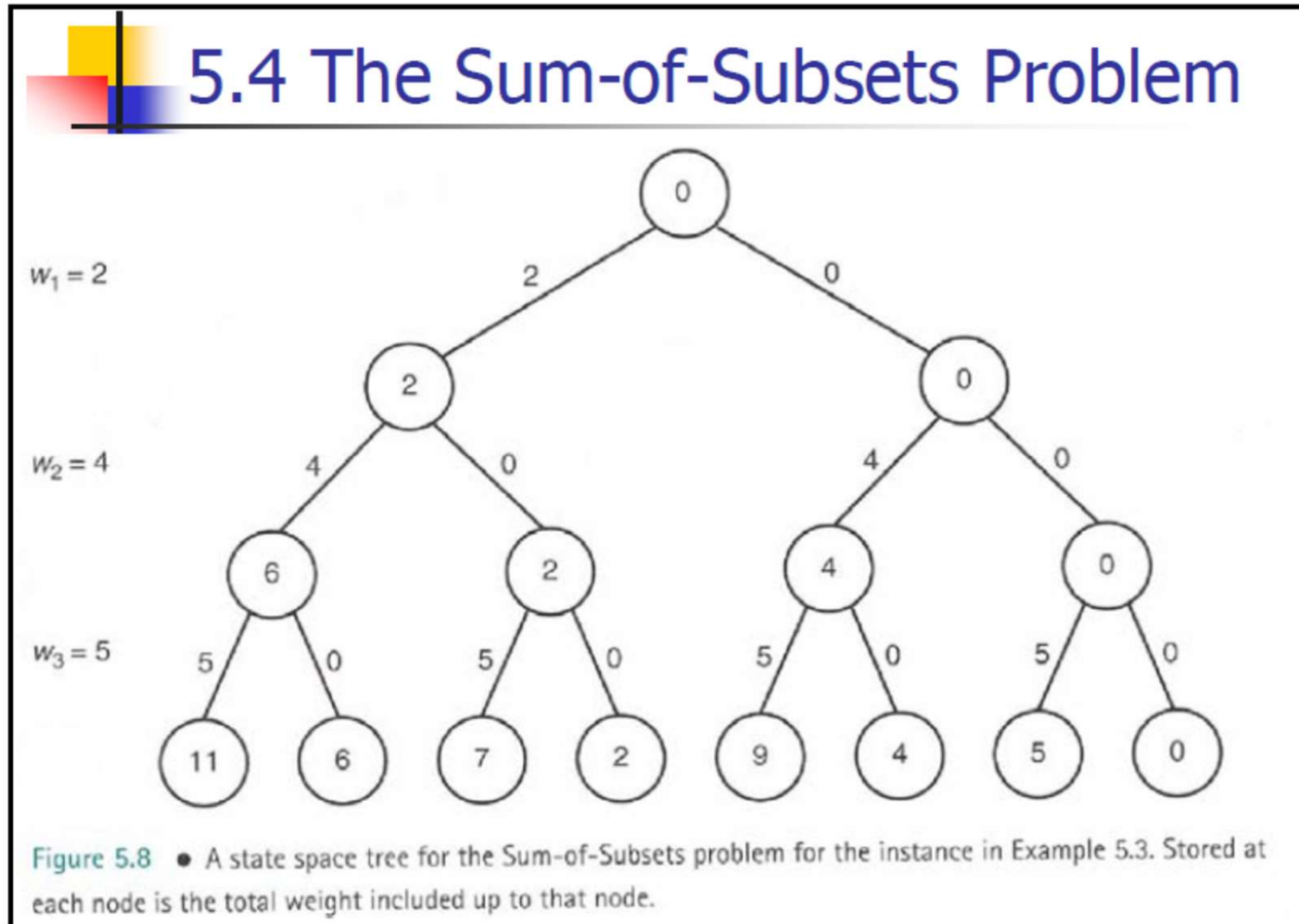
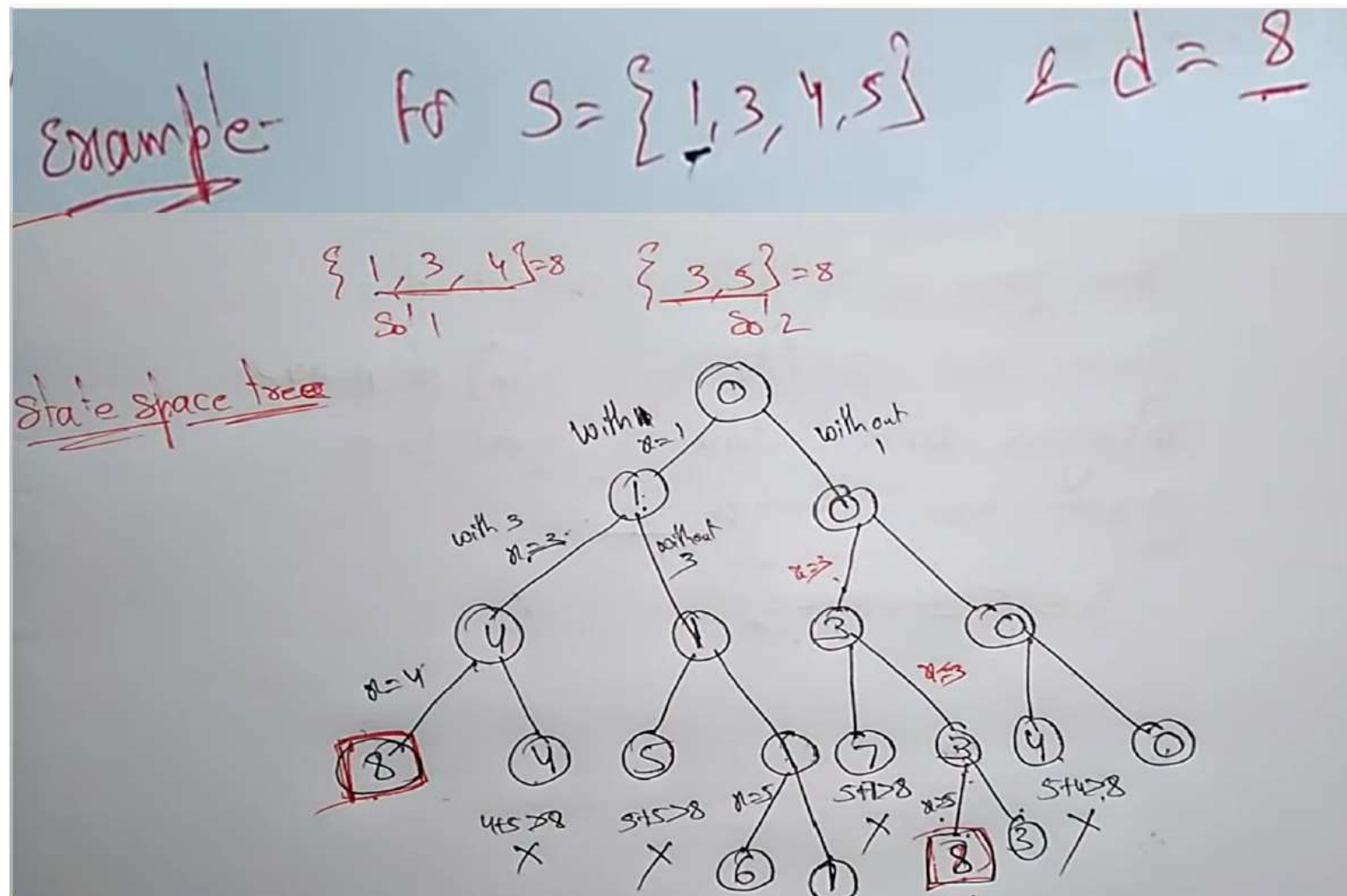


Figure 5.7 • A state space tree for instances of the Sum-of-Subsets problem in which  $n = 3$ .

# State space tree

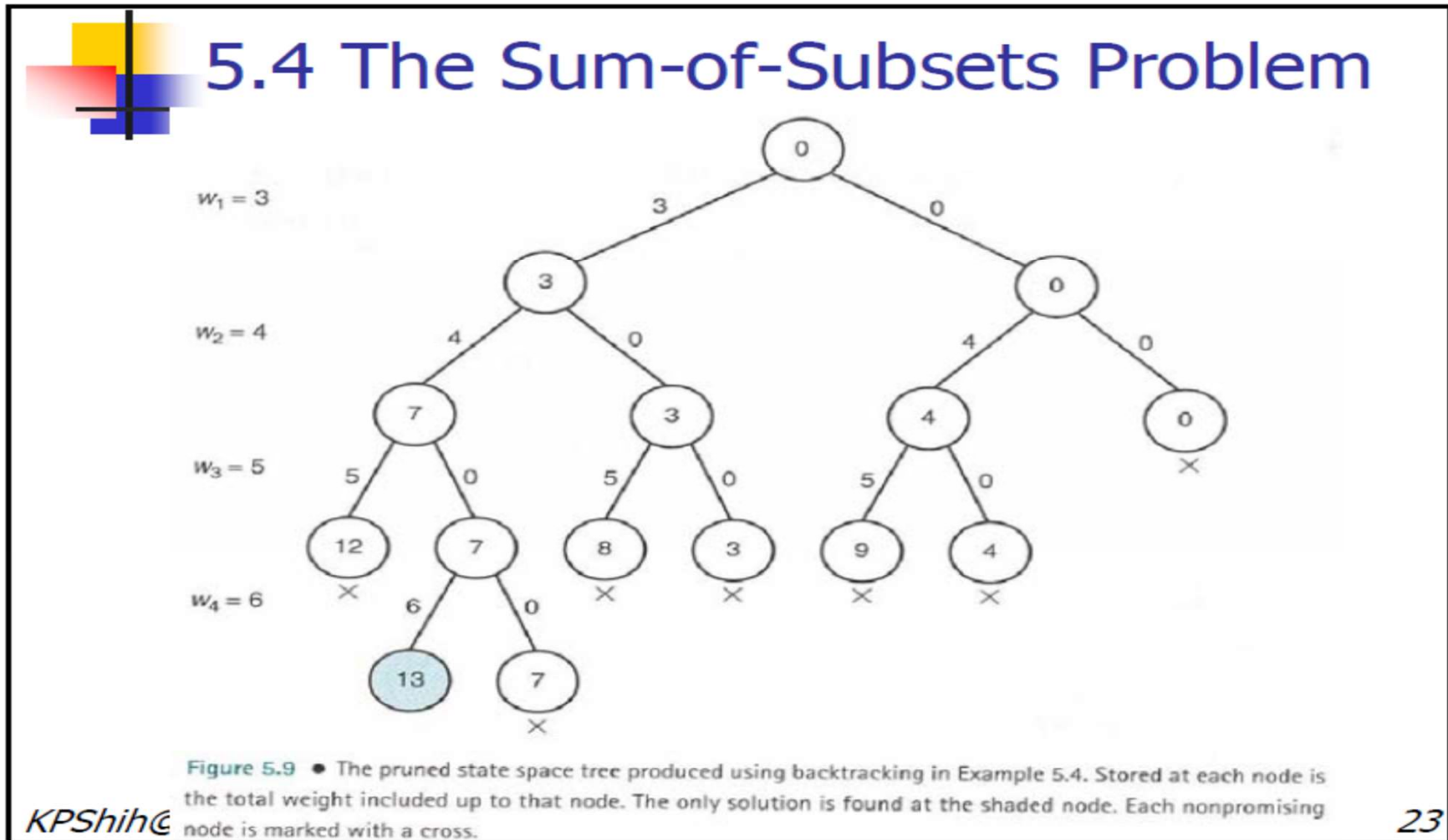


# Example-1



# Example-2

## State space tree where $W = 13$



```
void subset_sum(int list[], int sum, int starting_index, int target_sum)
{
    if( target_sum == sum )
    {
        subset_count++;
        if(starting_index < list.length)
            subset_sum(list, sum - list[starting_index-1], starting_index, target_sum);
    }
    else
    {
        for( int i = starting_index; i < list.length; i++ )
        {
            subset_sum(list, sum + list[i], i + 1, target_sum);
        }
    }
}
```

- $O(2^N)$  time complexity