**19CSE302 – Design and Analysis of algorithms**

**Assignment – 19.11.2021**

**Name:** R.Abhinav.

**Roll:** CB.EN.U4CSE19453

1. **Coin Change Problem:**

**Basic Goal of the problem**:

* Given, an array of coin values and an amount that needs to be returned we have to find out the minimum number coins by which we can achieve that amount.

**Inputs:**

* we get the array of all the coins available we get the amount that needs to be fullfilled using the coins above

**Basic Assumption we make:**

* we have infinite number of each coins that is given in the array.

**Branch and Bound approach:**

**Logic :**

* Here we nearly try out all possible combinations of giving change to the given amount. We keep track of the minimum possible coins required till a particular instance and try to ignore the possibilities where the coins required exceed the min possible coins required at that instance. To use this constraint to its fullest, the search order should be in the descending order of the denominations.

**Implementation:**

minCoins = float("inf")  
import time  
  
  
def BranchAndBound(arr, n, count):  
 global minCoins  
 if n < 0 or count > minCoins:  
 return  
 if n == 0:  
 minCoins = min(minCoins, count)  
 return  
 for i in arr:  
 BranchAndBound(arr, n - i, count + 1)  
  
  
def main():  
 denominations = list(map(int, input().split()))  
 n = int(input())  
 start\_time = time.time()  
 denominations.sort(reverse=True)  
 BranchAndBound(denominations, n, 0)  
 print("Answer = ", minCoins)  
 print("Execution Time = ", (time.time() - start\_time))  
  
  
main()

**Testcase - 1:**

* [ 1 , 2, 5 , 10 , 20 , 50 , 100 , 200 , 500 , 2000 ] → Coins
* 2750 → amount

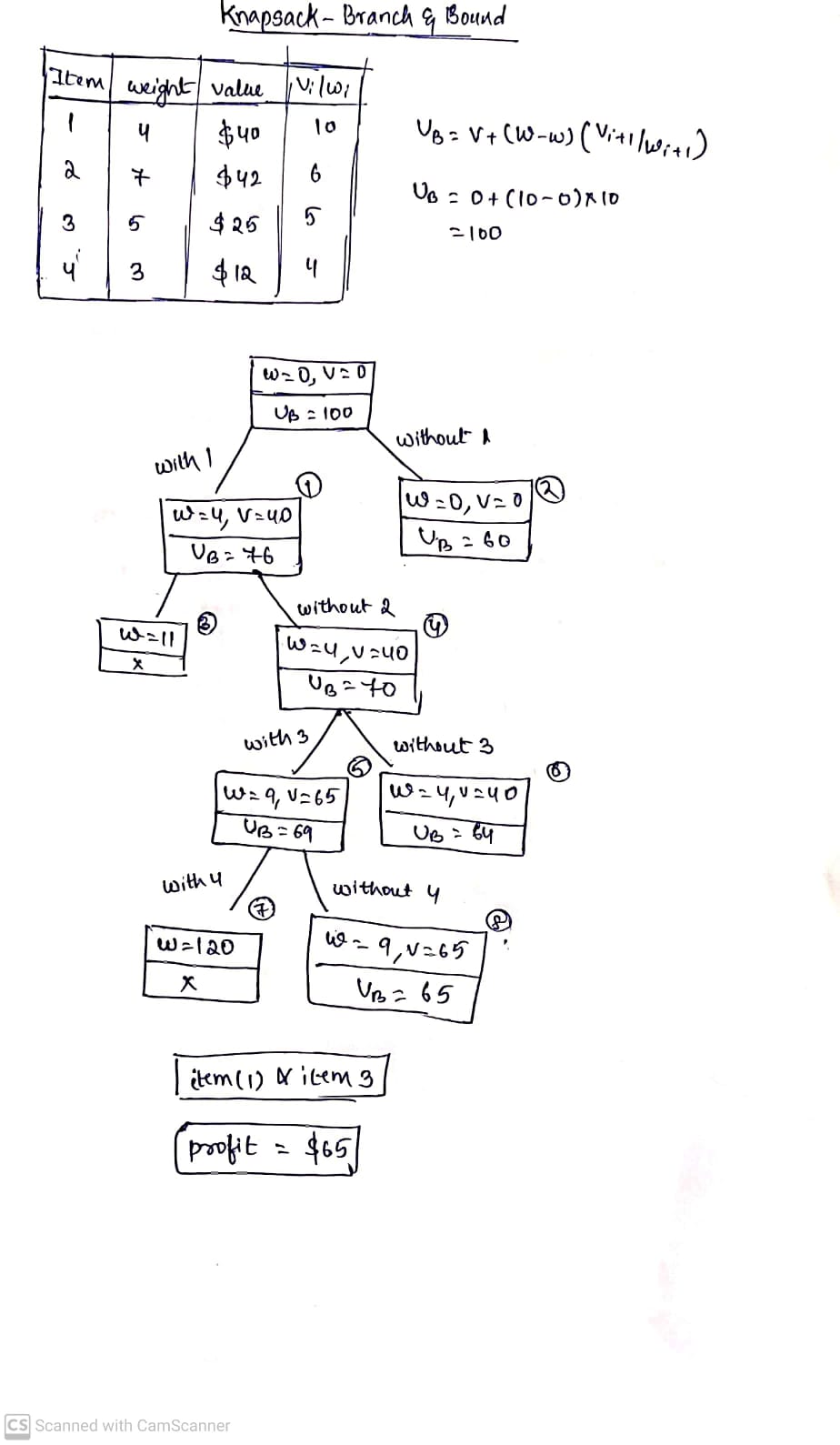
**Expected output:** 2000 + 500 + 200 + 50 = 2750 → 4

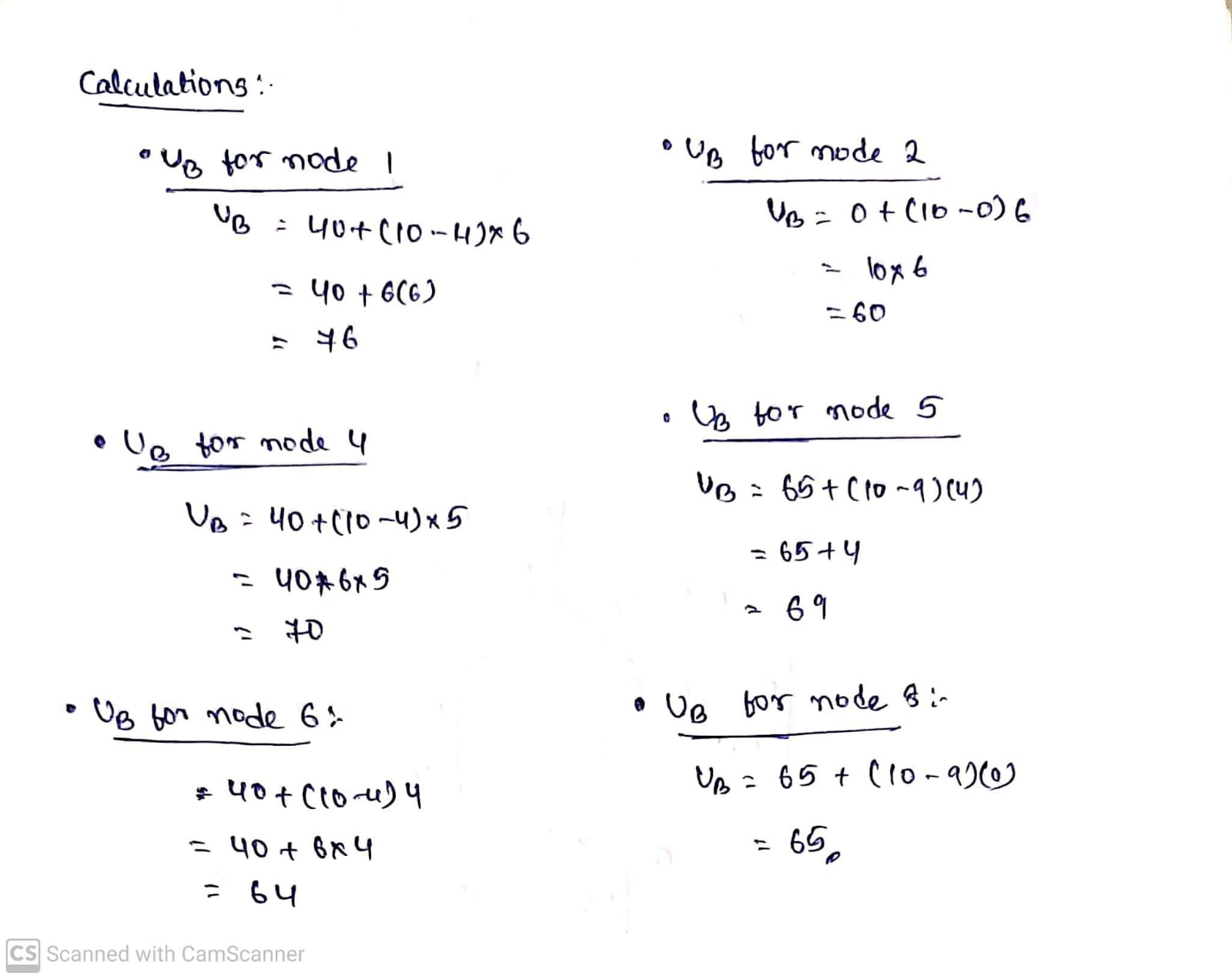
**Output :** 4

****

**Correctness:** As the algorithm takes in all possible ways excluding the ways that are definitely costlier and gives more number of coins, it will always be the accurate solution

1. **Knapsack:**

****

****

**Code :**