0/1 KNAPSACK PROBLEM USING BB

Pseudocode:

```
Algorithm:
BB_KNAPSACK(cp, cw, k)
BEGIN
 cp ← 0
 cw ← 0
 k ← 1
 IF (cw + w[k] ≤ M) THEN
  Y[k]  ← 1
  IF (k < n) THEN
     BB_KNAPSACK(cp + p[k], cw + w[k],
     k + 1)
  END IF
  IF (cp + p[k] > fp) && (k == n) THEN
     fp ← cp + c[k]
     fw ← cw + w[k]
     X ← Y
  END IF
 END IF
 IF BOUND(cp, cw, k) ≥ fp THEN
  Y[k] ← 0
  IF (k < n) THEN
     BB_KNAPSACK(cp, cw, k + 1)
  END IF
  IF( cp>fp ) && ( k == n ) THEN
     fp ← cp
     fw ← cw
     X ← Y
  END IF
 END IF
END FUNCTION
```

```
BOUND(cp, cw, k)
BEGIN
 b ← cp
 c ← cw
 FOR i ← k + 1 to n DO
   IF (c + w[i] ≤ M) THEN
     c ← c + w[i]
     b ← b – p[i]
   END
 END
 RETURN b
END FUNCTION
```

Time complexity:

- Even though this method is more efficient than the other solutions to this problem, its worst case time complexity is still given by $O(2^n)$, in cases where the entire tree has to be explored.
- However, in its best case, only one path through the tree will have to explored, and hence its best case time complexity is given by O(n).