(Q7)

```
// B[B1, B2, ..., Bn] array of books B.
// W given combined width
// solveMinHeight() gives the minimum total height after
   arranging the books as per given rules

def solveMinHeight(index, B, W, Wleft, Hmax):
    if index > B.length:
        return Hmax
    currHeight = B[index].height
    currWidth = B[index].width
    if (currWidth <= Wleft):
        existingShelf = solveMinHeight(index+1, B, W,
                            Wleft - currWidth, max(Hmax, currHeight))

    newShelf = Hmax + solveMinHeight(index+1, B, W,
                            W - currWidth, currHeight)

    return min(newShelf, existingShelf)
```

Note : We are assuming that each element of array B has two components, named as B[i].height and B[i].width representing the height and width of each book respectively.

Explaination:

Each book can be placed in 2 ways—

① on the current existing shelf

② on the new next shelf

After the current book B[index] is placed we can move on to place other books similarly by using same recursive calls.

consider case 1:

In this case the remaining width of the total width W must be greater than the existing book's width (i.e., the book we are currently trying to place). We must send the updated height and width of this current shelf recursively calling the next book, i.e, remaining width can be updated to (W - currWidth) and height can be updated as max(Hmax, currHeight).

consider case 2:

We are placing the B[index] book in the new shelf. To place it we will make a call irrespective of any condition to explore all possible options of arranging books. Now as we are moving to a new shelf we must add Hmax of present shelf as in future recursive calls it will be lost, thus we are adding it in line no. 7. Here again we must send updated height and ~~wid~~ width of new shelf as currHeight and (W - currWidth) respectively.

Thus for each book we are exploring all possibilities, (though maintaing the given order of the books), by looking for if a book can be placed in the current shelf or we need to make a new shelf. We require the minimum height of the book shelf such that books can be placed following the given constraints, so we will return min(newShelf, existingShelf).

Analysis

$T(i)$ = time complexity of solveMinHeight()

$$T(i) = 2T(i+1) + O(1)$$

Let, $n - i = i' \Rightarrow i = n - i'$

$$T(n-i') = 2T(n-i'+1) + c$$

Let, $S(i') = 2S(i'-1) + c$

$$= 2\{2S(i'-2) + c\} + c$$

$$= 2^2 S(i'-2) + c + 2c$$

$$= 2^2\{2S(i'-3) + c\} + c + 2c$$

$$= 2^3 S(i'-3) + c + 2c + 2^2 c$$

$$\vdots$$

$$= 2^k S(i'-k) + (c + 2c + 2^2 c + \cdots + 2^{k-1} c)$$

Let $i' - k = 1$

$$S(i') = 2^k S(1) + c * \frac{2^k - 1}{2 - 1}$$

$$= 2^k * 1 + c * (2^k - 1)$$

$$S(i') \cong O(2^{i'-1})$$

$$T(1) = S(n-1) \le S(n) \quad \text{(assuming } i = 1\text{)}$$

$$S(n) = O(2^{n-1}) \cong O(2^n)$$

Ans.

$$S(n) = O(2^n)$$