1.

1. Sort by $W_i + S_i$ decreasing and if same by W_i by decreasing

   We can use counting sort for this and cost $O(N + W_{max} + S_{max})$
2. Remove unimportant item

   If the occurence of one item with the weight W_i is above $\frac{W_{max} + S_{max}}{W_i}$

   We don't need to consider that item again

   Using harmonic series, this will make the remaining item at most $S \lg S$

   $O(N)$
3. Use DP[i][j] which define by:

   The largest total weight with item from i to n with the available remaining weight possible is j.

   $O((S_{max} + W_{max}) \cdot \lg(S_{max} + W_{max}) \cdot S_{max}) = O((S_{max}^2 + S_{max} \cdot W_{max}) \cdot \lg(S_{max} + W_{max}))$
4. The base case happen when i = number of item

   If i = number of item, the largest weight is 0
5. $DP[i][j] = max(DP[i+1][j], W_i + DP[i+1][min(j - W_i, S_i)])$

   We need to make sure that for the latter case, $W_i \leq j$ is always happen
6. So the answer should be in DP[0][inf]

   In this case, inf = $W_{max} + S_{max}$
7. Backtrack the items

   Keep track for every DP state, whether they come from the path that pick item$_i$ or not

   $O(S_{max} + W_{max})$

Final Complexity:

$$= O(N + W_{max} + S_{max} + N + (S_{max}^2 + S_{max} \cdot W_{max}) \cdot \lg(S_{max} + W_{max}) + S_{max} + W_{max})$$
$$= O(N + W_{max} + S_{max} + (S_{max}^2 + S_{max} \cdot W_{max}) \cdot \lg(S_{max} + W_{max}))$$

‘