Solution.

(a) Let s be the total sum of the elements in S. If s is odd, then there is no solution. Otherwise, we reduce the problem to solving the SubsetSum with $T = \frac{s}{2}$ and S being unchanged. This will take $O(n)$ reduction.

(b) If $s \leq 2T$, introduce a new element in S such that the sum of all elements in S, say s, plus the new element t equals $2T$. With this new set, solve Partition. The new element t would belong to one of the equal sets, if they exist, so the other set would represent the subset of the original list that satisfies the conditions.

In the case where $s > 2T$, we must consider two cases based on the parity of s.

- If s is even, then solve the Partition. The partitions will be of size $T + x$ for some residue x, and so we simply need to identify if x is in S. If it is, then we have solved our problem.
- If s is odd, check to see if the sum of all the odd integers in S, say s_{odd}, meets the conditions $s_{\text{odd}} \leq T$ and $T - s_{\text{odd}}$ is even (0 is even). If so, remove all odd integers from S and reattempt the algorithm with $T_{\text{new}} = T - s_{\text{odd}}$. If no sufficient partition was found, then the solution cannot contain all of the odd integers, so now consider then removing each odd integer from S and applying the logic from the even case, as the total sum will now be even.

The worst case is when $s > 2T$ and s is odd, which would require $O(n^2)$ operations, but it can be optimised by calculating the total sum by subtracting the remove element from the original total.