1.

Solution:

My Algorithm:

We first sort A is decreasing order, and then re-label the numbers from the biggest to the smallest as $\{a_1, a_2, a_3,..., a_n\}$ where a_1 is the largest number.

After A is sorted we will build S by picking a subset of A (starting from the beginning/largest numbers), such that the sum of S is as large as it can be but less than B.

Algorithm:

merger-sort(A) in decreasing order
relabel A's elements

S = $\emptyset$ AND T = 0
FOR i = 1 to i = n DO
IF T + $a_i \leq$ B THEN
$S = S \bigcup a_i$
T = T + a_i
return S

Time Complexity:

If we use merge-sort we can do the sorting of A is O(nlogn).

And in the loop we are looping thought n elements which makes the complexity O(n) with primitive oporations inside the loop.

The overall complexity is O(nlogn).

Proof:

I'll use proof by contradiction here.

Lets assume we get a set S from the algorithm such that the sum of its elements is less than half of the maximal sum of the a feasible solution S'.

This means that we at least have one a_i in S' that we don't have in S.

From the algorithm we have, this means that the addition of a_i to S would mean that the sum of it's elements with a_i would become larger that B. Since the elements in A are ordered in decreasing order, a_i can only be as large as the last element added to S (a_{i-1}). This means that:

1. The sum of the elements in S is already larger than B/2 and adding a_i will push it to be larger than B.

2. That $a_i >$ B/2.

If we are in case 1, since B is the largest that the sum of the elements of S' can be, then if sum of S is B/2 then we have proven the algorithm will return the set S we want.

If we are in case 2, then since the elements that we take from A and add to S are in decreasing order. Then the elements taken before a_i are $\geq$ that it, or $\geq$ than B/2 which means that the sum of elements in S is $\geq B/2$. Which also proves that the algorithm give the desired S.

In both cases we will get an S such that the sum of it's elements are greater than or equal to B/2 or half the maximal sum of any feasible solution.

Resource: http://homepage.cs.uiowa.edu/ sriram/3330/fall15/