Question 3,

Select any two elements from set A[], to make x1+x2 = target,
If exist, return true,
Else, return false.
Pseudo, we need to sort the set A first,
We use merge sort to sort the elements in set A
Using the pseudo code in textbook

```
Merge–sort (A,p,r)
If p<r
q = [(p+r)/2]
merge-sort (A,p,q)
merge-sort (A,q+1,r)
merge(A,p,q,r)
```

the running time for merge sort is $\Theta(n\lg(n))$
after sorting the set A, we use the following algorithm to tell whether or not there exist two elements in A whose sum is X

```
int i = 0
int j = n
while i<j do:
        if A[i]+A[j] == X:
                return true
        if A[i]+A[j] <x:
                i++;
                continue
        if A[i]+A[j]>x:
                j--;
                continue
end while
return false
```

the time complexity for the above code is $\Theta(n)$, cause in worst case, we only need to iterate all the element in array A, and spend constant time on each element to tell whether or not the sum of them is equals to the target value. So the total running time for these two part (first part is merge sort, the second part is above pseudo code) is dominated by the sort part which is $\Theta(n\lg(n))$.
Generally, the first step is merge sort time is $t1 = \Theta(n\lg(n))$
The second step is find the sum, using two "pointer", the first one is in the start of the array, the second one is in the end of the array, running time is $t2 = \Theta(n)$
So $T(n)= t1 + t2= \Theta(n\lg(n))$