1. SUM(S, x)

```
MergeSort(S, I, length[S]) #sort S with mergesort
low = 1
high = length[S]
while low < high
    if x == S[low] + S[high]
        then return true #if found return true
    elseif x > S[low] + S[high]
        then low = low + 1 #update low
    else
        high = high – 1 #update high
return false #return false if no sum is found
```

2. To do stack operations with push, pop and min in constant O (1) time, we are going must implement it by using two stacks, actual and auxiliary, where auxiliary stack will always be the minimum element.

Push	Pop
if stack is empty, insert x top of actual and auxiliary	pop the top element of actual stack
If stack is not empty • Let y be top of auxiliary. If x <= y, then push x to auxiliary and actual • Ley y be top of auxiliary. If x >= y, then push x to actual only	If top element of actual equals auxiliary stack, then also pop top element of auxiliary.