Q2

```
// let the set of n fractions be A
// let the size of the set be n
Q2_iter(A,n)
Boundary_max = min(A)    //O(n)
E = 1
While 1 == 1 do
    sum _of_A = 0
    while counter < n do
        sum_of_A += x[counter]/(y[counter] - E)
    end do
    If sum_of_A == S do
        break
    If sum_of_A > S do
        E = (Boundary_max - E) / 2
    else do
        E =  (Boundary_max + E) / 2
Return E
```

Time complexity: n(find min) + n*log(set(min))
Therefore the time complexity is O(n * log(min(set))