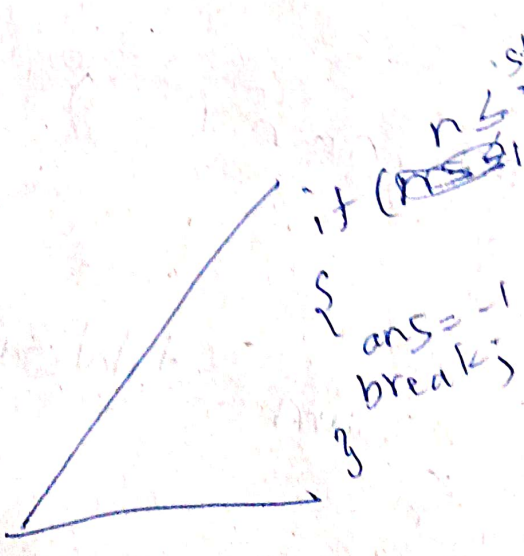
```
91) MoM(Array, l, n, k) {
      m = n/5
      For(i = 1 to m)
          M[i] = median(Array, 5i-4, 5i);
    Median ← MoM(M, 1, m, m/2)
    r ← Partition(A, I, n, median)
  if (k < r)
        return MoM(Arr I, r-1, k);
  else if (k > r)
        return MoM(Array, r+1, n, k-r);
  return median

int main( )
{
   .  .  .
   Sum = total sum of all weights
}
```

```
int temp1=0, temp2=0, start=1
while (true)
{
  median = MoM(Array, start, n, n/2)
  for (int i=start; i<=n; i++)
  {
    if (Array[i] < tempmedian)
      temp1 += weight[i]
    else if (Array[i] > median)
      temp2 += weight[i]
  }
  if (temp1 < sum/2 && temp2 < sum/2)
  { ans = median
    break;
  }
  else if (temp1 > sum/2)
  { n = n/2
    temp2 = 0
  }
  else
  { start = n/2
    temp2 = 0
  }
}
```



My logic in words

(1) Finding the median by MoM

(2) checking if median satisfies the given condition condition
if yes median is the ans

(3) if No is $W(S_{<x}) > \frac{W(S)}{2}$
the ans will be less than the current median, so applying same logic for the left half of array

(4) else applying same logic for right half

Time complexicity

worst case = $cn + n + \frac{cn}{2} + \frac{n}{2} + \frac{cn}{4} + \frac{n}{4} + \ldots$

cn ↓ MoM, n ↓ calculating sums

$= (c+1)n\left(1 + \frac{1}{2} + \frac{1}{4} + \ldots\right)$

$= (c+1)n\left(\frac{1}{1-\frac{1}{2}}\right)$

$= 2(c+1)n$

$= O(n)$ ✓