

Isaac Abboudi

DetectTerrorist write-up

February 19 2023

In my algorithm, we split the group of passengers in half and weigh each side. The side with the lower weight must include the fake tefillin so we discard the rest of the passengers and continue with the group in question. If the group is an odd number of people, we remove one person from the group -set them to the side- and continue; and when we encounter the next odd numbered group we bring that person back to even things out. When we get to the end and we have a group of size 2, we return both values and any odd-man-out that might still be saved and check between the three which is lightest.

In terms of correctness, consider that with each divide there are only two options, right is lighter or left is lighter and whichever is heavier gets discarded - we know the terrorist cannot be on that side by definition of the weight discrepancy. As we reduce the input we get to a sub array of 2, in which case we get out of the recursion and compare the 2 options left -and the third if applicable.

The recurrence relation is  $T_n = T(n/2) + c$  - c being the constant Weigh and other constant value-checking statements. The closed form is  $T_n = O(\log n)$  and since as a binary search we cut the input in half at each step the log of the base is 2.

I wrote a recursive method to be called with the constructor, which would set the terrorist value to the correct index. When getTerroist() is called, the result is returned without calculation.

My doubling ratio experiments came out to 1.45-2.02. Practically, my experiments won't conform to the closed form of the recurrence because in our recurrence we assume the weighing is constant yet in the actual program it is not.