Algorithms. Assignment 5

Problem 9

The following type of problem can appear in biological testing, in communication protocols, and some other fields. But do not worry about applications. For the sake of this exercise you can just accept the problem in this abstract form:

We are given a set S of n physical objects. Exactly one object in S is "special", but we do not know which one. We have the possibility to test arbitrary subsets $R \subseteq S$. The test outcome is binary: A positive test outcome tells us that the special object is in R. A negative test outcome tells us that the special object is not in R (and hence in $S \setminus R$).

(a) Give a test strategy that identifies the special object by using a minimal worst-case number of tests. How many tests do you need? Note that the question is about the exact number of tests, not about O-notation.

(b) Is your worst-case test number in (a) optimal? If so, explain why.
If not, re-work (a).

Hopefully you have immediately recognized the similarity to a known simple problem and applied divide-and-conquer. But now a new difficulty comes up: For technical reasons it is not possible to test large sets R. Specifically, every test set R is allowed to contain at most k objects, where k is some known fixed number (which might be much smaller than n). After a moment of thinking, the following test strategy naturally comes to mind:

Test disjoint subsets of S, each of size k, until some set R is tested positively. In this event, search for the special object in R, as in (a). (Since we have already $|R| \leq k$, the size restriction is irrelevant from now on.) If all tests are negative so far, and less than $2k$ objects remain, then search among the remaining objects, as in (a), as well.

So this is a nice combination of a greedy rule and divide-and-conquer. But we have good reasons to be skeptical about greedy rules. Maybe it is better to test less than k objects in every step, because this makes the final search phase in the positive set R faster?

However, the proposed strategy is in fact optimal. We use two little intermediate steps to prove this. (See next page.)

Let $T_k(n)$ be defined as the worst-case number of tests needed, if $|S| = n$, and tests are limited to at most k objects.

(c) We claim that

$$T_k(n) = 1 + \min_{j \le k} \max\{T_k(j), T_k(n-j)\}.$$

Explain why this is true.

Just in case, the notation means: For any fixed n and k, and for any j, take the maximum of the two terms in the curly brackets, finally take the minimum of these maxima, for all j that are less than or equal to k.

(d) We claim that T_k is a monotone function, that is: For all $i \le j$ we have $T_k(i) \le T_k(j)$. Explain why this is true (which should be very easy).

(e) Finally, prove optimality of the proposed test algorithm. It might be a good idea to use the results of (c) and (d).

If you cannot manage the entire problem, it is OK to submit what you have, and ask for advice about points where you struggle.