TIN093 Algorithms - Assignment 5

Mirco Ghadri

September 1, 2024

Problem 9

a) Our test strategy selects a subset R which has a size of $\lceil n/2 \rceil$ in case n is not an even number. If the test is positive, we repeat the recursive algorithm on R, else we repeat this recursive algorithm on $S \setminus R$. The number of tests is equal to $\lceil log_2(n) \rceil$.

b) The worst-case number of tests is optimal. The reason is that we want to minimize the number of remaining items to search for after each test. So we consider the maximum number of remaining items in the worst case, i.e $\max\{k, n-k\}$. Why? Because lets always assume the item is in the larger subset(since we do worst-case analysis), so we have to search there every time. So we select the $k \leq n$ that minimizes this maximum value, i.e the maximum number of remaining items. Let x denote this value.

$$x = \min_{k \leq n} \max\{k, n-k\} \tag{1}$$

To find the value of k that minimizes $x = \max\{k, n-k\}$, we proceed as follows:
Let $k_2 = n - k$. We have:

$$k + k_2 = n$$

Therefore:

$$n = k + k_2 \leq 2 \cdot k_{\text{max}}$$

This implies:

$$\frac{n}{2} \leq k_{\text{max}}$$

which sets a lower bound for k_{max}.
The smallest possible value for k_{max} is achieved when:

$$k_{\text{max}} = \frac{n}{2}$$

Thus, the value of k that minimizes x is:

$$k = \frac{n}{2}$$

Again, if n is uneven, we do $k_{max} = \lceil \frac{n}{2} \rceil$

c) The formula

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

gives us the minimum number of tests we need to do in the worst-case. Why? First of all, each subset size j is less than or equal to k. The number of tests is equal to 1 + the number of tests performed on the subset. We select the largest subset(since we analyze worst-case) but we select the j that minimizes the largest subset each time. Hence we get the minimum number of total tests we need to perform in the worst case scenario where the item is always in the largest subset.

d) To prove that $T_k(i)$ is a monotone function, lets assume we have $x \le y$ and we want to prove that $T_k(x) \le T_k(y)$. We have:

$$T_k(x) = 1 + \min_{j \le k} \max\{T_k(j), T_k(x-j)\} = 1 + T_k(x') \tag{2}$$

We also have:

$$T_k(y) = 1 + \min_{j \le k} \max\{T_k(j), T_k(x-j)\} = 1 + T_k(y') \tag{3}$$

We claim that:

$$x \le y \implies x' \le y' \tag{4}$$

Why? Because for any value of $j \le k$ we have:

$$x \le y \implies x - j \le y - j \tag{5}$$

Hence, we have that:

$$x \le y \implies \max\{j, x-j\} \le \max\{j, y-j\} \tag{6}$$

So we know that x' will always be less than or equal to y'. Why is this significant? It is significant because it means that $T_k(x)$ will converge at least as fast if not faster than $T_k(y)$, which means that $T_k(x) \le T_k(y)$ since in each iteration/test, we add 1 to the total number of tests.

e) The value of j that minimizes the largest T_k value is indeed $j = k$. When the number of tests t are less than or equal to $2k$, the value of j that minimizes the largest T_k value is $\frac{t}{2}$ as proven in b). Since T_k is a monotone function as proven in d), minimizing it, means minimizing the total number of tests.