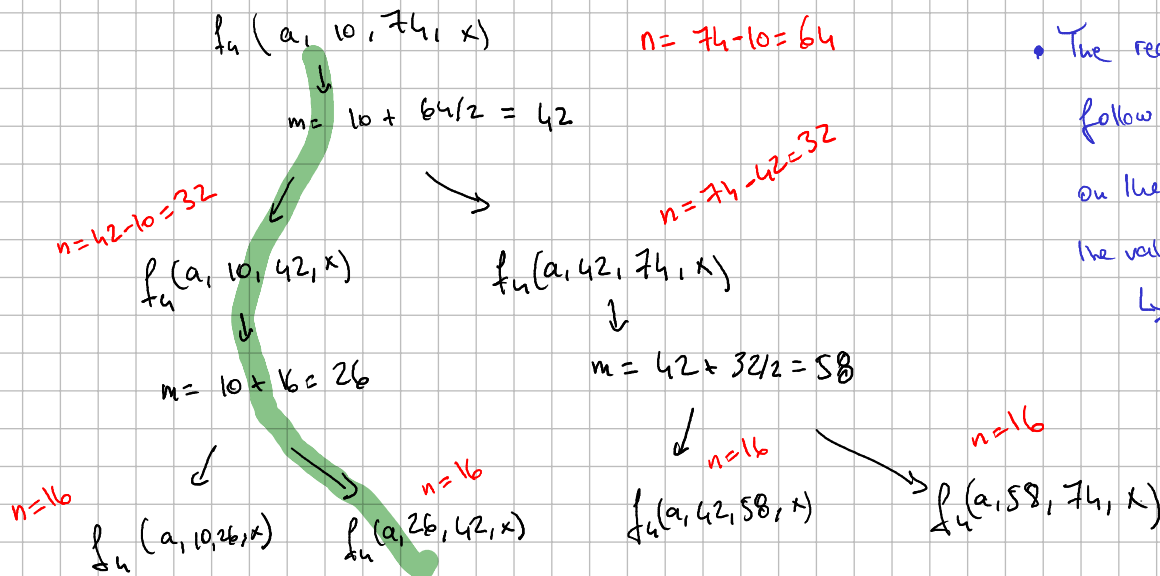


```

// Size of the problem is  $n = hi - lo$ .
// You can assume  $n = 2^p$  for some integer  $p$ .
int f4(int a[], int lo, int hi, int x){
    if(lo >= hi){
        return -1;
    }
    int m = lo + (hi - lo)/2;
    if(a[m] < x){
        return f4(a, lo, m, x);
    }else{
        return f4(a, m, hi, x);
    }
}

```

suppose $lo = 10$ and $hi = 74$, $n = 74 - 10 = 64$.



- The recursive execution will follow one path depending on the value of x and on the values stored in a .
 ↳ for example the green path to the left.

- At each level of the recursion, a constant amount of work is performed: a pair of tests, a fixed number of arithmetic integer operations and assignments, and a call/return.
- There are $\log_2(n)$ recursive calls.

The tree has $2n-1$ nodes.

$$\begin{aligned}
 & \log_2(n) + 1 \\
 & \underbrace{1 + 2 + 4 + \dots + n}_{2^0 \quad 2^1 \quad \dots \quad 2^{\log_2 n}} = 2^{\log_2 n + 1} - 1 \\
 & = 2n - 1
 \end{aligned}$$

↳ f_4 is therefore in $\Theta(\log_2 n)$

regardless of the values stored in a and of the value of x .