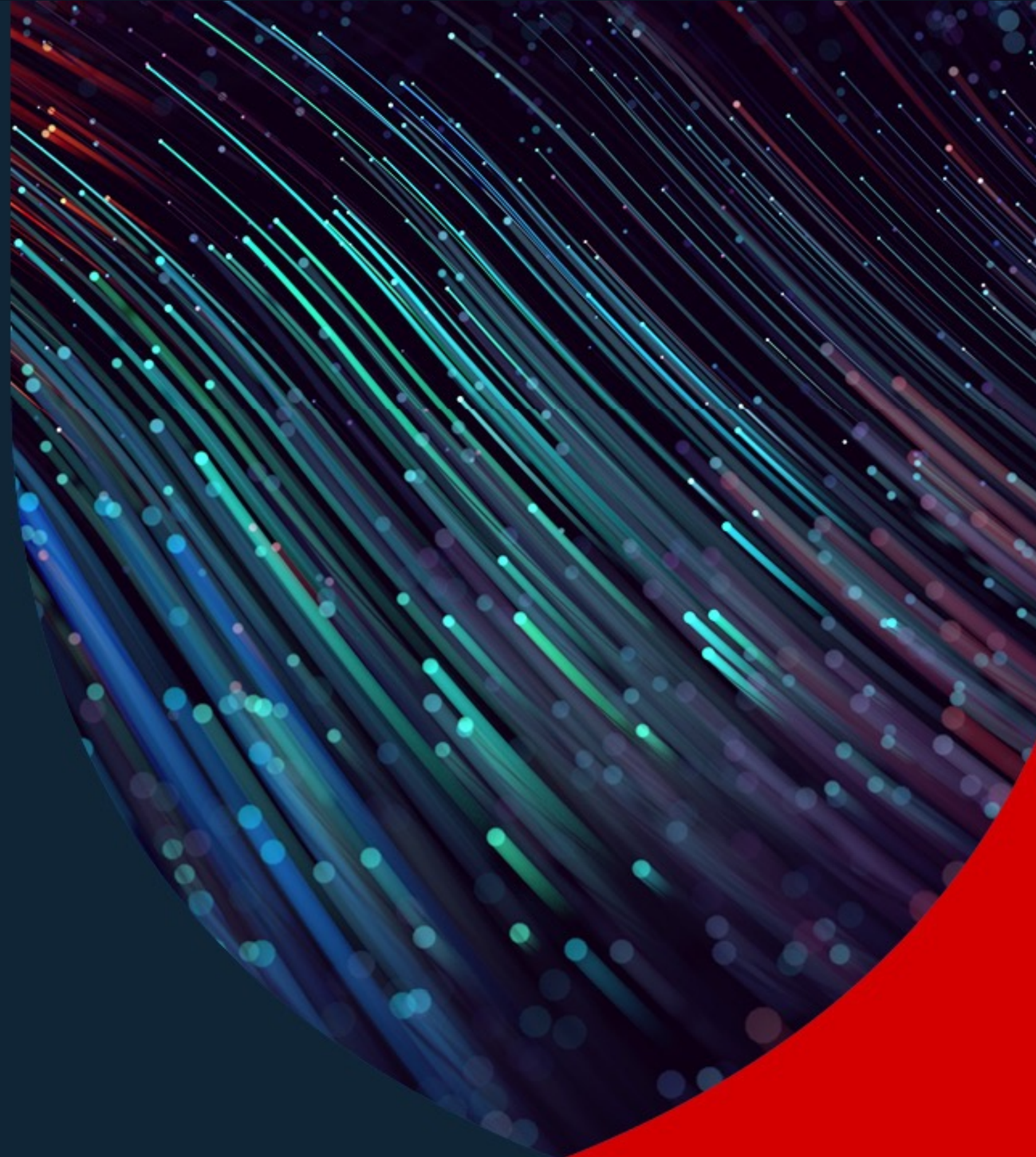
make history.

THE UNIVERSITY of ADELAIDE

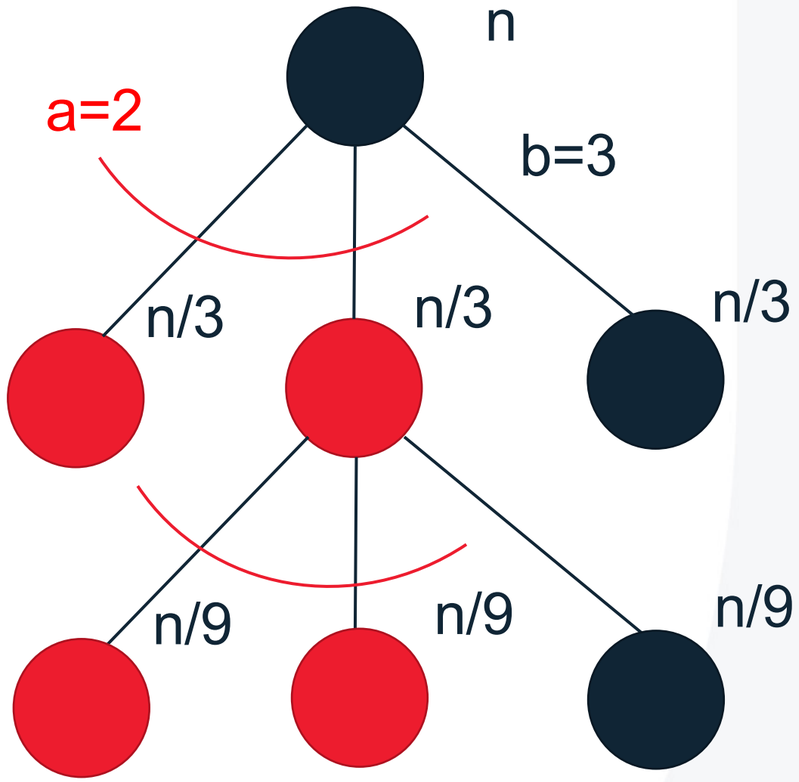
Master theorem

Dr. Anna Kalenkova



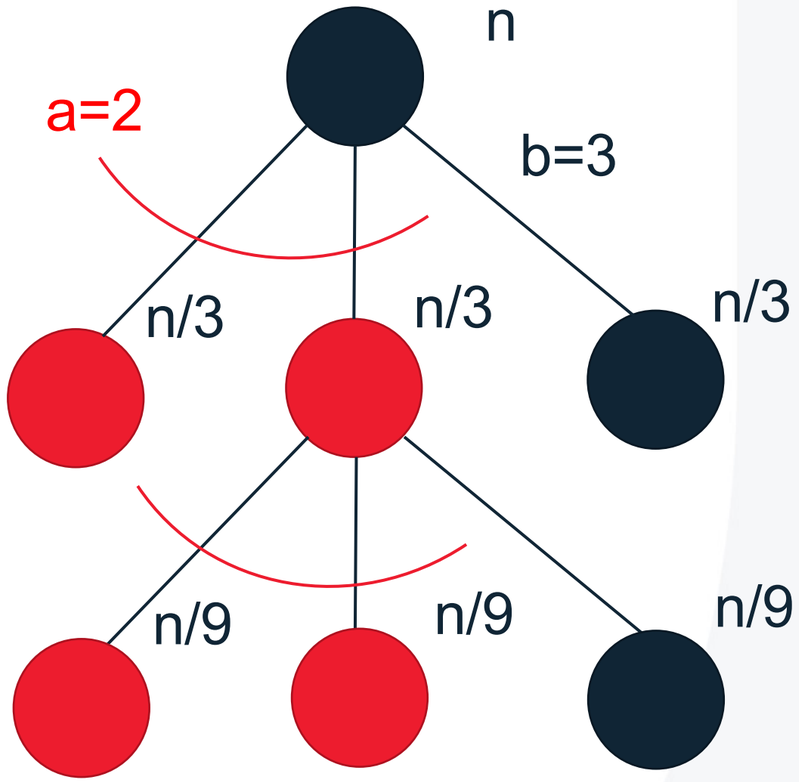
Divide and conquer

- Create *a* subproblems, each having size *n*/*b*
- Call the procedure recursively on each subproblem
- Combine the results from the subproblems



Master theorem

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$



T(n) – computational complexity to solve problem of size n;

f(n) – computational complexity to combine results from subproblems.

THE UNIVERSITY of ADELAIDE

Master theorem

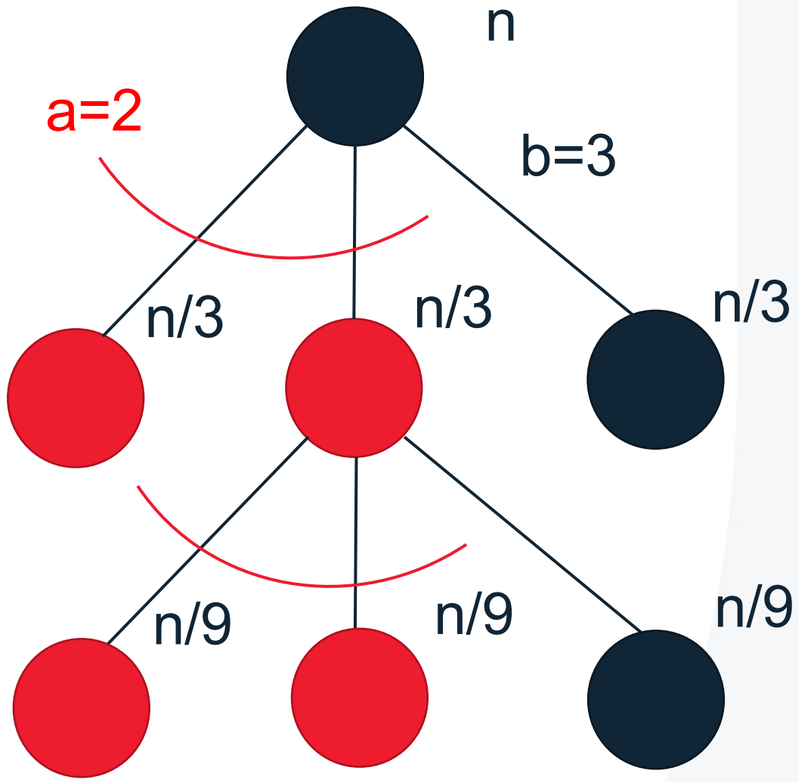
For constants $a \geq 1, b \geq 2, d \geq 0$ and $f(n) \in \Theta(nd)$,

consider the recurrence:

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

then

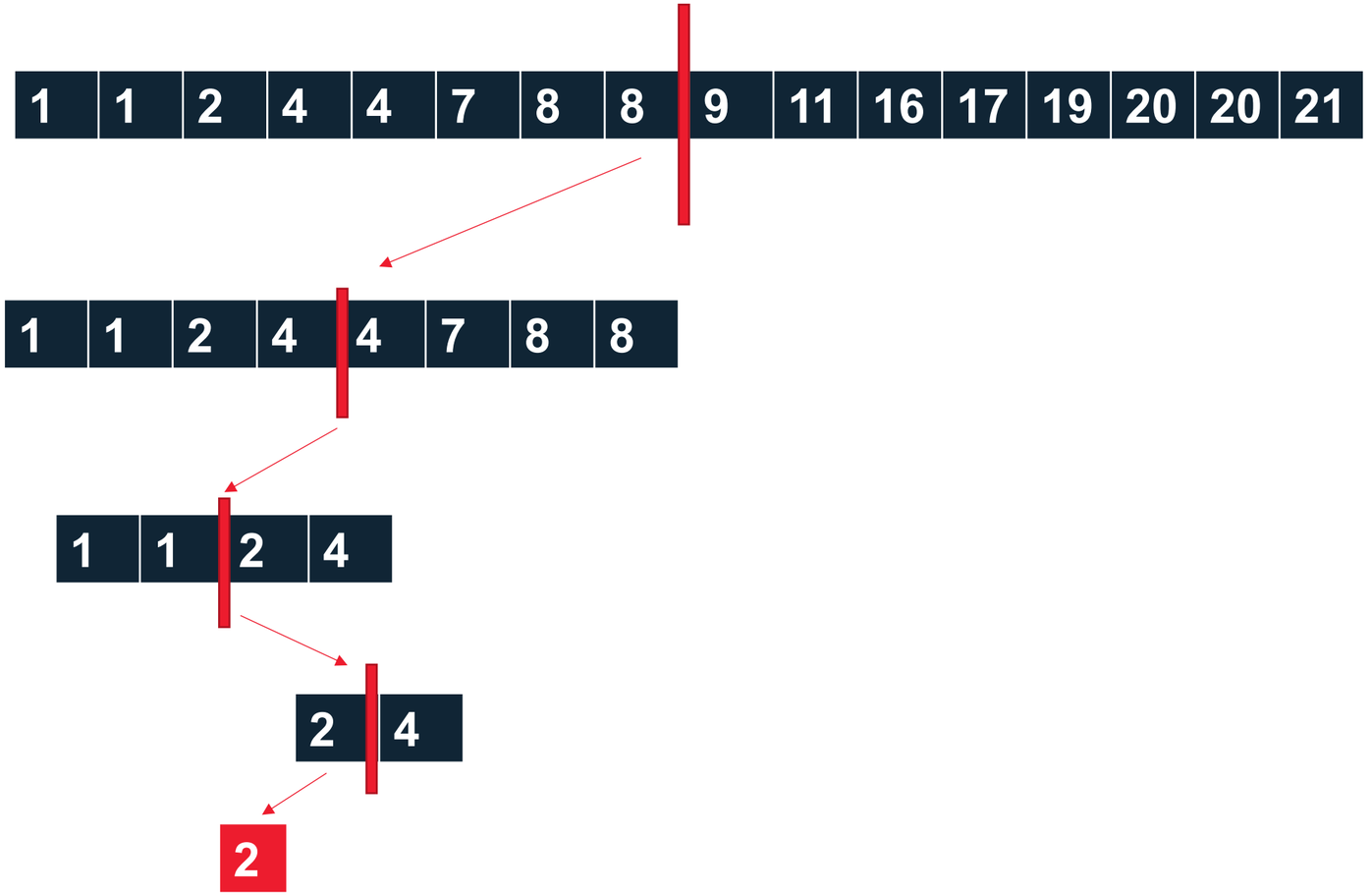
$$\mathrm{T}(n) \in \begin{cases} \Theta(n^d), & \text{if } a < b^d \\ \Theta(n^d \log n), & \text{if } a = b^d \\ \Theta\left(n^{\log_b a}\right), & \text{if } a > b^d \end{cases}$$



THE UNIVERSITY of ADELAIDE

Binary search

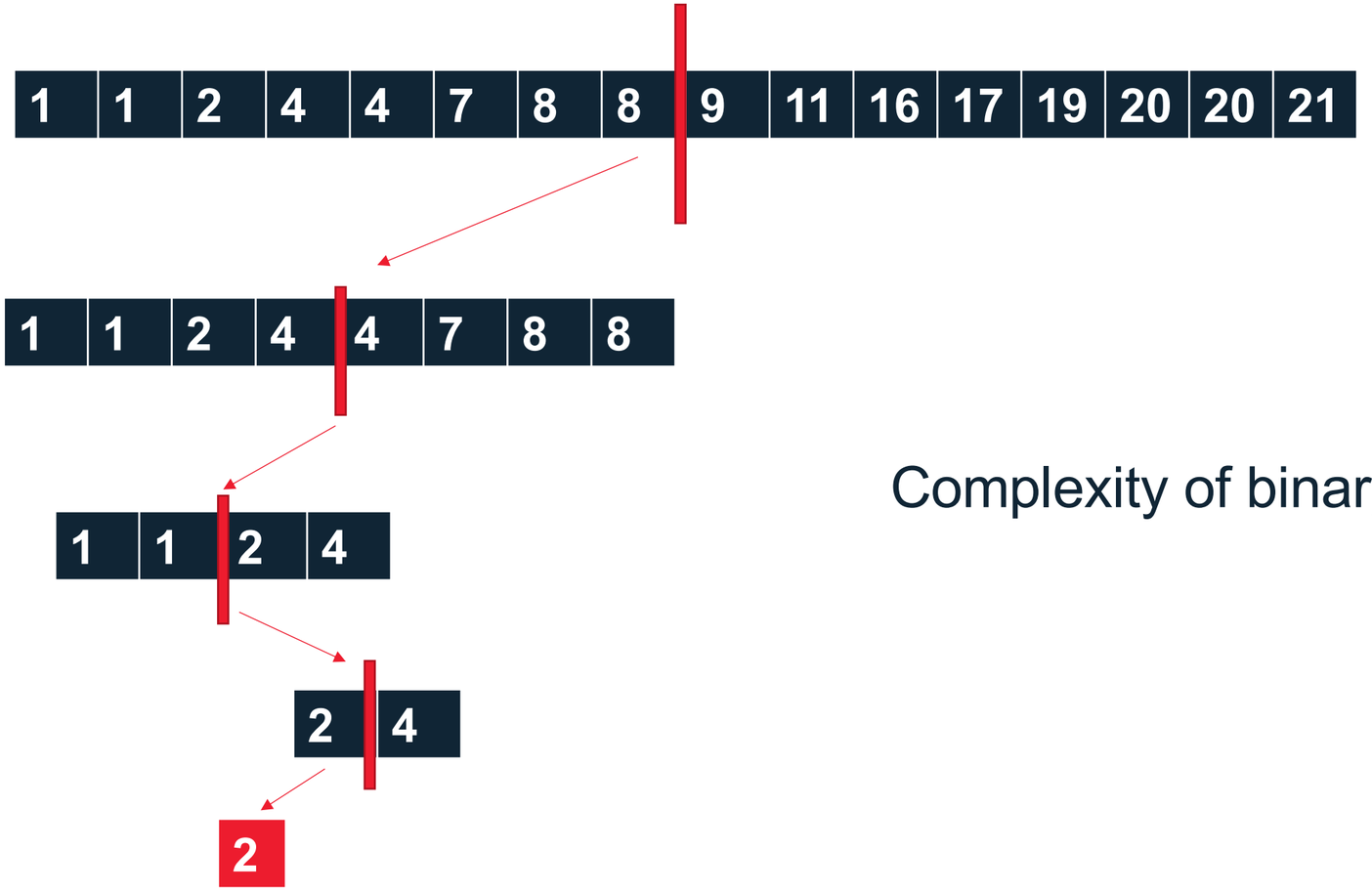
Suppose we need to find element 2 in the sorted array.



THE UNIVERSITY of ADELAIDE

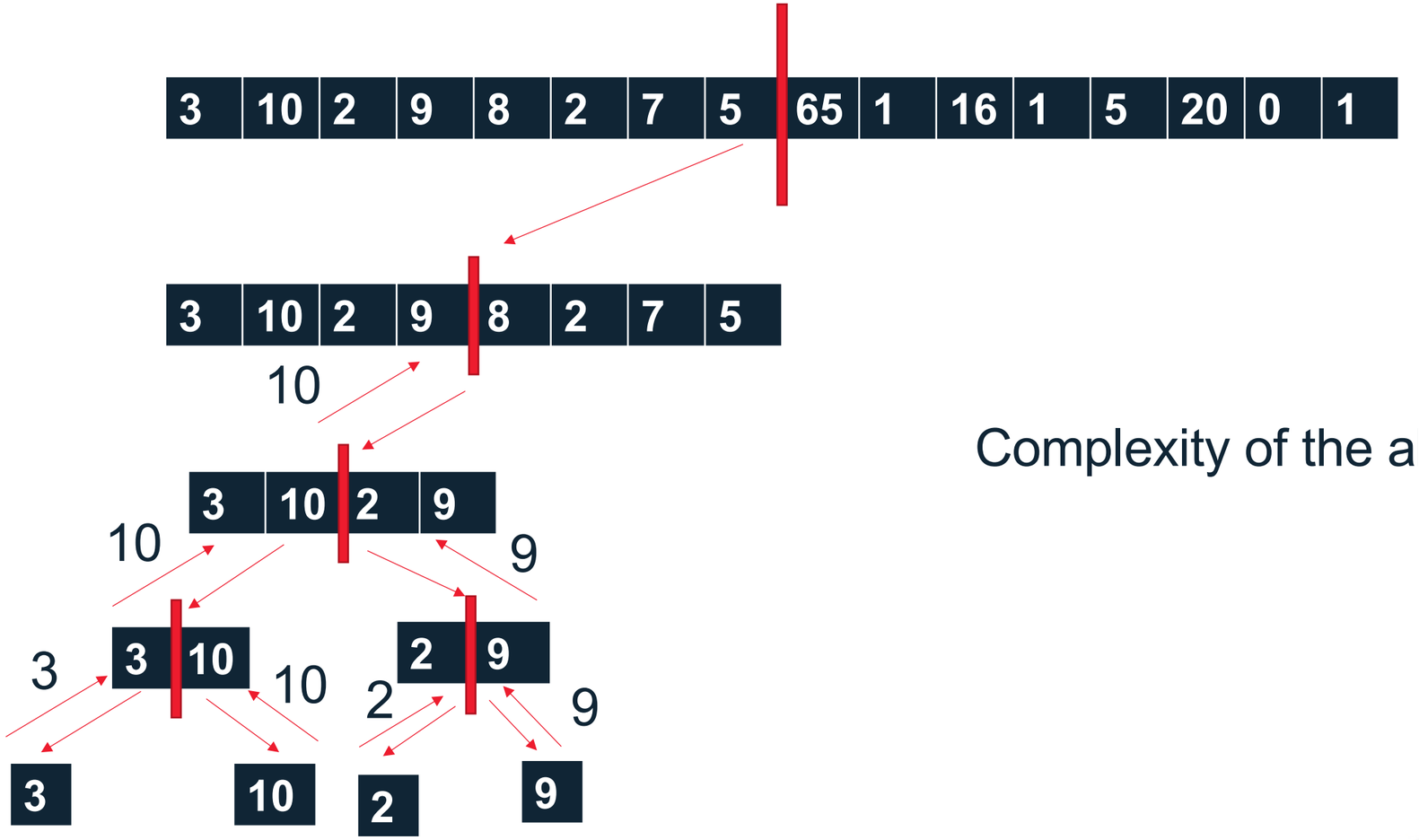
Binary search

Suppose we need to find element 2 in the sorted array.



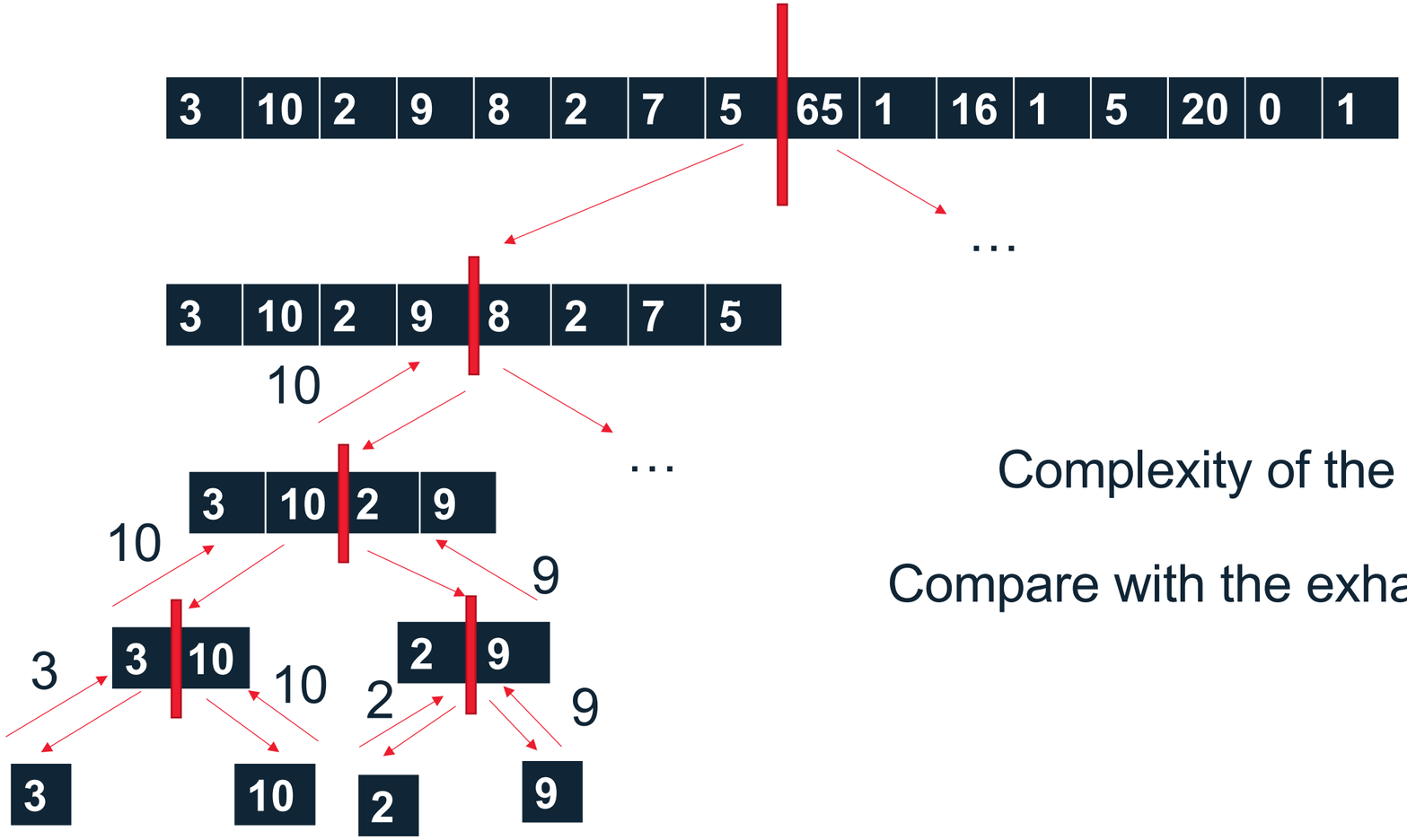
Complexity of binary search?

Find max elements



Complexity of the algorithm?

Find max elements



Complexity of the algorithm?

Compare with the exhaustive max search algorithm

THE UNIVERSITY of ADELAIDE