
# DESIGN AND ANALYSIS OF ALGORITHMS

**Surabhi Narayan**

Department of Computer Science & Engineering

# DESIGN AND ANALYSIS OF ALGORITHMS

## Transform and Conquer

**Surabhi Narayan**
Department of Computer Science & Engineering

Transform and Conquer:

- In Transformation stage, the problem's instance is modified to be, for one reason or another, more amenable to solution.
- Then, in the second or conquering stage, it is solved.

There are three major variations

- Transformation to a simpler or more convenient instance of the same problem—we call it ***instance simplification***.
- Transformation to a different representation of the same instance—we call it ***representation change***.
- Transformation to an instance of a different problem for which an algorithm is already available—we call it ***problem reduction***.

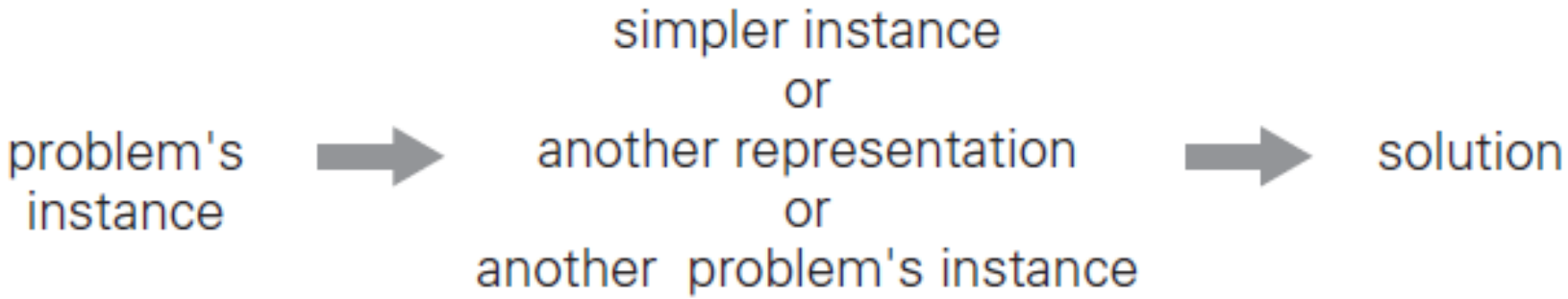


**FIGURE 6.1** Transform-and-conquer strategy.

*Checking element uniqueness in an array*

The brute-force algorithm compared pairs of the array's elements until either two equal elements were found or no more pairs were left. Its worst-case efficiency was in $(n^2)$.

Alternatively, we can sort the array first and then check only its consecutive elements: if the array has equal elements, a pair of them must be next to each other, and vice versa.

**ALGORITHM** *PresortElementUniqueness*$(A[0..n-1])$

//Solves the element uniqueness problem by sorting the array first
//Input: An array $A[0..n-1]$ of orderable elements
//Output: Returns "true" if $A$ has no equal elements, "false" otherwise

sort the array $A$

**for** $i \leftarrow 0$ **to** $n-2$ **do**
**if** $A[i] = A[i+1]$ **return false**
**return true**

$$T(n) = T_{sort}(n) + T_{scan}(n) \in \Theta(n \log n) + \Theta(n) = \Theta(n \log n).$$


# THANK YOU

**Surabhi Narayan**
Department of Computer Science &Engineering

**surabhinarayan@pes.edu**