
Programming Design In-class Practices

Algorithms and Recursion

Ling-Chieh Kung

Department of Information Management
National Taiwan University

Complexity

- Let's watch a **video**!

Problem 1: linear search

- Given a size- n array A of integers and a target integer t , check whether t is in A .
- Please design a function that:
 - Takes an integer array A , its length n , and an integer t as parameters.
 - Returns true if t is in A and false otherwise.
- Please write **pseudocode** by writing a **loop**.

Problem 2: linear search (recursion)

- Given a size- n array A of integers and a target integer t , check whether t is in A .
- Please design a function that:
 - Takes an integer array A , its length n , and an integer t as parameters.
 - Returns true if t is in A and false otherwise.
- Please write **pseudocode** with **recursion**.

Problem 3: Hanoi Tower

- Given an integer n , the fastest way to solve the Hanoi Tower problem is unique.
- Let's watch a **video**!
- Though you have read the code solving the Hanoi Tower problem, please still write **pseudocode** for it.

Problem 4: insertion sort

- Let's watch a [video](#)!
- Idea: Given a size- n integer array A whose first k elements are sorted (from small to large) and the remaining is not, insert the $(k + 1)$ th element into the first half to make the first $k + 1$ element sorted.

```
insertionSort(a non-repetitive array  $A$ , the array length  $n$ )  
  for  $i$  from 1 to  $n$   
    //  $A_{1..(i-1)}$  is sorted  
    insert  $A_i$  to the proper place within  $A_{1..(i-1)}$   
    // now  $A_{1..i}$  is sorted
```

- Please refine the pseudocode to make it [precise](#).
 - Make you (and your friend) know how to execute each step.