

**01076564 Design and Analysis of Algorithms Majority Voting**

**เสนอ**

**ผศ.ดร.ชุติเมศฎ์ ศรีนิลทา**

**จัดทำโดย**

**นายพงศธร สิริธัญกุล 56010778**

**นายรพีพัชร สุวะศรี 56010999**

**ภาคเรียนที่ 2 ปีการศึกษา 2556**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์**

**คำนำ**

รายงานนี้เป็นส่วนหนึ่งของวิชา Design and Analysis of Algorithm มีเนื้อหาเกี่ยวกับ Unique Integers in an Array จัดทำขึ้นเพื่อแสดงผลการคิดและวิเคราห์ทางแนวคิดของอัลกอลิทึมตามกำหนดเพื่อแก้ปัญหาดังกล่าว

ผู้จัดทำหวังเป็นอย่างยิ่งว่าเนื้อหาภายในรายงาน จะสามารถแสดงแนวการคิดของผู้จัดทำแก่ผู้ที่สนใจได้อย่างมีประสิทธิภาพ หากผิดพลาดประการใดต้องขออภัยมา ณ ที่นี้ด้วย

**ผู้จัดทำ**

**สารบัญ**

[Unique Integers in an Array - General Algorithm Report 4](#_Toc481542717)

[อัลกอลิทึมการแก้ปัญหาที่ใช้ 4](#_Toc481542718)

[แนวคิดเบื้องต้น 4](#_Toc481542719)

[Pseudo code of an algorithm 6](#_Toc481542720)

[Analyze the theoretical run-time complexity of the algorithm. 7](#_Toc481542721)

[Discuss the correctness of algorithm. 8](#_Toc481542722)

[Simulation Screenshot 9](#_Toc481542723)

# Unique Integers in an Array - General Algorithm Report

Design and Analysis of Algorithm

## อัลกอลิทึมการแก้ปัญหาที่ใช้

### แนวคิดเบื้องต้น

มีสอง array เป็น array ของ input และของ output โดยทำการประมวลผลค่าของ array input แล้วบันทึกว่าเก็บอะไรไว้บ้าง ก่อนหน้าจะนำไปเก็บที่ array output ให้ตรวจสอบว่าเคยมีการบันทึกไว้หรือยัง **ถ้าไม่** ให้เพิ่มเข้า array ผลลัพธ์ **ถ้าใช้** ไม่ต้องทำอะไร เนื่องจากแนวคิดหลักมีเพียงเท่านี้ ความต่างของความเร็วจะขึ้นกับวิธีการค้นหาข้อมูล

เป็นการค้นหาอย่างง่ายโดยสร้าง array ขึ้นมาขนาดเท่าค่าสูงสุดของเลขที่มีแล้วให้แต่ละตำแหน่งของ array สัมพันธ์โดยตรงกับค่าของสมาชิกของ input

**เมื่อทำการค้นหาตัวเลข *i*** จะตรวจสอบ array ตำแหน่งที่สัมพันธ์กับ *i*  
**หากค่าเป็นจริง** ถือว่าเจอ   
**หากค่าไม่เป็นจริง** ถือว่าไม่เจอ

**ตัวอย่าง** [4 2 6 4 3 5 5]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iterative | INPUT | CHECKER | OUTPUT | Action |
| 0 |  | [F F F F F F] | [] |  |
| 1 | 4 | [F F F T F F] | [4] | PUSH |
| 2 | 2 | [F T F T F F] | [4 2] | PUSH |
| 3 | 6 | [F T F T F T] | [4 2 6] | PUSH |
| 4 | 4 | [F T F T F T] | [4 2 6] | IGNORE |
| 5 | 3 | [F T T T F T] | [4 2 6 3] | PUSH |
| 6 | 5 | [F T T T T T] | [4 2 6 3 5] | PUSH |
| 7 | 5 | [F T T T T T] | [4 2 6 3 5] | IGNORE |

**ตัวอย่าง** [1 2 3 3 4 3 3]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iterative | INPUT | CHECKER | OUTPUT | Action |
| 0 |  | [F F F F] | [] |  |
| 1 | 1 | [T F F F] | [1] | PUSH |
| 2 | 2 | [T T F F] | [1 2] | PUSH |
| 3 | 3 | [T T T F] | [1 2 3] | PUSH |
| 4 | 3 | [T T T F] | [1 2 3] | IGNORE |
| 5 | 4 | [T T T T] | [1 2 3 4] | PUSH |
| 6 | 3 | [T T T T] | [1 2 3 4] | IGNORE |
| 7 | 3 | [T T T T] | [1 2 3 4] | IGNORE |

## Pseudo code of an algorithm

Define [inputs] and [outputs] as an (Array).

“Accepts all input into &[inputs]”.

Define [check array] as an (Array of /false/).

Define method “Find if [index] is exists”.

If [check array.index] is /true/.

Return /true/.

Else

Return /false/.

Define method “Set [index] as exists”.

Set [check array.index] to be /true/.

For each member of [inputs], name it as [i].

“Find if [i] is exists” and save result as [exists].

If exists is not /true/.

With [outputs] “Push [i] as member”.

“Set [i] as exists”.

For each member of [outputs], name it as [i].

“Display [i + ’, ’]”.

## Analyze the theoretical run-time complexity of the algorithm.

“Find if [index] is exists” --- F

“Set [index] as exists”. --- S

Main Process

## Discuss the correctness of algorithm.

เนื่องจากการแนวคิดเป็นเรื่องของการตรวจสอบว่าเคยมีสิ่งใดสิ่งหนึ่งมาก่อนหรือเปล่า แนวคิดในการตรวจสอบว่ามีอยู่หรือเปล่าอย่างง่ายที่สุดคือการทำรายการของที่เคยมีมาแล้ว

อย่างไรก็ตามรายการอย่างง่ายในแนวคิดที่ได้แสดงไปได้ใช้ข้อจำกัดตามโจทย์นั่นคือเป็นเลขจำนวนเต็ม และวิธีการเชื่อมโยงตำแหน่งกับค่าของเลขไม่ได้กำหนดไว้ตายตัว จึงใช้กับเลขจำนวนลบได้ อย่างไรก็ตาม การใช้วิธีดังกล่าวสิ้นเปลืองหน่วยความจำมาก หากตัวเลขมีการกระจายที่มาก

## Simulation Screenshot



