**Data Structures and Algorithm**

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

**การทดลองที่ 8 :** เปรียบเทียบการทำงานของ sequential search และ binary search

**จุดประสงค์**

1. นักศึกษาเข้าใจการทำงานและผลลัพธ์ของ sequential search และ binary search พร้อมทั้งสามารถเปรียบเทียบได้
2. นักศึกษาเข้าใจข้อจำกัดของ sequential search และ binary search ที่นำไปใช้ในงานต่างๆ

**โปรแกรมตัวอย่าง**

|  |
| --- |
| import random  comparecount = 0  def binary\_search(arr, low, high, x):  global comparecount  if high >= low:  mid = (high + low) // 2  if arr[mid] == x:  return mid  elif arr[mid] > x:  comparecount +=1  print("comparecount = ",comparecount, "low = ",low, "high =",mid-1)  return binary\_search(arr, low, mid - 1, x)  else:  comparecount +=1  print("comparecount = ",comparecount, "low = ",mid+1, "high =",high)  return binary\_search(arr, mid + 1 , high, x)  else:  return -1  def sequential\_search(arr,x):  global comparecount  for i in arr:  comparecount +=1  if i==x:  return comparecount  return -1  datcount = 100000  ## incase of sequential search  #arr = [random.randint(1,10000000) for i in range(datcount)]  ## in case of binary search  #arr = sorted([random.randint(1,10000000) for i in range(datcount)])  ## in case of succesfully search  #x = arr[random.randint(1,datcount)]  ## in case of unsuccessfully search  #x = -1  ## worstcase successfully search  #x = arr[-1]  print("key =",x)  print("data len = ",len(arr))  ## in case of binary search  #result = binary\_search(arr, 0, len(arr)-1, x)  ## incase of sequential search  #result = sequential\_search(arr,x)  print("compare count = " ,comparecount)  if result != -1:  print("Element is present at index", str(result))  else:  print("Element is not present in array") |

จากโปรแกรมตัวอย่าง เป็นโปรแกรมสำหรับการทดสอบ sequential search และ binary search ใน 3 กรณี คือกรณีที่ค้นหาเจอแบบทั่วไป กรณีที่ค้นหาเจอแบบแย่ที่สุด และกรณีที่ค้นหาไม่เจอข้อมูลที่ต้องการ โดยจะต้องมีการปรับแต่งโปรแกรมโดย uncomment บรรทัดที่ต้องใช้งานให้ถูกต้อง โดยจุดสำคัญคือข้อมูลที่ใช้สำหรับ binary search ต้องเป็นข้อมูลที่ทำการเรียงลำดับแล้ว และข้อมูลที่ใช้สำหรับ sequential search ไม่จำเป็นต้องเรียงลำดับก็ได้

สำหรับการทดลองนี้จะใช้ขนาดข้อมูลคงที่คือ 100,000 ชุดข้อมูล และใช้การทดลองซ้ำๆ เพื่อหาค่าเฉลี่ยของการทำงานพื้นฐาน โดยนับที่จำนวนของการเปรียบเทียบข้อมูลอ้างอิง กับข้อมูลที่ต้องการค้นหา หากมีจำนวนการเปรียบเทียบน้อยกว่าจะถือว่ามีการทำงานที่รวดเร็วกว่า

**ตอนที่ 1 : การทำงานของ Sequential Search**

**Successfully Search , Average Case :**

1. ให้นักศึกษาทดลองโปรแกรมที่กำหนดให้ โดยกำหนดให้การทำงานเป็น sequential search ที่ข้อมูล 100,000 ชุดข้อมูล โดยกำหนด key ที่จะค้นหาเป็นแบบสุ่มตำแหน่งให้สามารถค้นหาเจอ
2. ทำการทดลองรันโปรแกรมทั้งหมด 10 ครั้งแล้วหาค่าเฉลี่ยของจำนวนการค้นหาทั้งหมด

|  |  |
| --- | --- |
| **ครั้งที่** | **จำนวนครั้งที่เทียบข้อมูล** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| **ค่าเฉลี่ย** |  |

ค่าเฉลี่ย / จำนวนชุดข้อมูลทั้งหมด =

**Successfully Search , Worst Case :**

1. ให้นักศึกษาทดลองโปรแกรมที่กำหนดให้ โดยกำหนดให้การทำงานเป็น sequential search ที่ข้อมูล 100,000 ชุดข้อมูล โดยกำหนด key ที่จะค้นหาเป็นข้อมูลตัวสุดท้ายของรายการ
2. ทำการทดลองรันโปรแกรมทั้งหมด 10 ครั้งแล้วหาค่าเฉลี่ยของจำนวนการค้นหาทั้งหมด

|  |  |
| --- | --- |
| **ครั้งที่** | **จำนวนครั้งที่เทียบข้อมูล** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| **ค่าเฉลี่ย** |  |

ค่าเฉลี่ย / จำนวนชุดข้อมูลทั้งหมด =

**Unsuccessfully Search :**

1. ให้นักศึกษาทดลองโปรแกรมที่กำหนดให้ โดยกำหนดให้การทำงานเป็น sequential search ที่ข้อมูล 100,000 ชุดข้อมูล โดยกำหนด key ที่จะค้นหาเป็นข้อมูลที่ไม่อยู่ในรายการ
2. ทำการทดลองรันโปรแกรมทั้งหมด 10 ครั้งแล้วหาค่าเฉลี่ยของจำนวนการค้นหาทั้งหมด

|  |  |
| --- | --- |
| **ครั้งที่** | **จำนวนครั้งที่เทียบข้อมูล** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| **ค่าเฉลี่ย** |  |

ค่าเฉลี่ย / จำนวนชุดข้อมูลทั้งหมด =

**ตอนที่ 2 : การทำงานของ Binary Search**

**Successfully Search , Average Case :**

1. ให้นักศึกษาทดลองโปรแกรมที่กำหนดให้ โดยกำหนดให้การทำงานเป็น binary search ที่ข้อมูล 100,000 ชุดข้อมูล โดยกำหนด key ที่จะค้นหาเป็นแบบสุ่มตำแหน่งให้สามารถค้นหาเจอ
2. ทำการทดลองรันโปรแกรมทั้งหมด 10 ครั้งแล้วหาค่าเฉลี่ยของจำนวนการค้นหาทั้งหมด

|  |  |
| --- | --- |
| **ครั้งที่** | **จำนวนครั้งที่เทียบข้อมูล** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| **ค่าเฉลี่ย** |  |

ค่าเฉลี่ย / จำนวนชุดข้อมูลทั้งหมด =

**Successfully Search , Worst Case :**

1. ให้นักศึกษาทดลองโปรแกรมที่กำหนดให้ โดยกำหนดให้การทำงานเป็น binary search ที่ข้อมูล 100,000 ชุดข้อมูล โดยกำหนด key ที่จะค้นหาเป็นข้อมูลตัวสุดท้ายของรายการ
2. ทำการทดลองรันโปรแกรมทั้งหมด 10 ครั้งแล้วหาค่าเฉลี่ยของจำนวนการค้นหาทั้งหมด

|  |  |
| --- | --- |
| **ครั้งที่** | **จำนวนครั้งที่เทียบข้อมูล** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| **ค่าเฉลี่ย** |  |

ค่าเฉลี่ย / จำนวนชุดข้อมูลทั้งหมด =

**Unsuccessfully Search :**

1. ให้นักศึกษาทดลองโปรแกรมที่กำหนดให้ โดยกำหนดให้การทำงานเป็น binary search ที่ข้อมูล 100,000 ชุดข้อมูล โดยกำหนด key ที่จะค้นหาเป็นข้อมูลที่ไม่อยู่ในรายการ
2. ทำการทดลองรันโปรแกรมทั้งหมด 10 ครั้งแล้วหาค่าเฉลี่ยของจำนวนการค้นหาทั้งหมด

|  |  |
| --- | --- |
| **ครั้งที่** | **จำนวนครั้งที่เทียบข้อมูล** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| **ค่าเฉลี่ย** |  |

ค่าเฉลี่ย / จำนวนชุดข้อมูลทั้งหมด =

**ตอนที่ 3 : ตอบคำถามและวิเคราะห์การทำงาน**

1. Binary search มีข้อจำกัดอย่างไรบ้าง

1. ถ้าเป็นข้อมูลที่ไม่มีการเรียงลำดับ ใน binary search นักศึกษาคิดว่าจะเกิดผลการทำงานเป็นอย่างไร

1. จากจำนวนข้อมูลที่เท่ากัน ใน worst case และกรณีที่ค้นหาไม่เจอ การทำงานของ search แบบไหนไวกว่ากัน

1. จากจำนวนข้อมูลที่เท่ากัน ในกรณีทั่วไป การทำงานของ search แบบไหนไวกว่ากัน

1. ในกรณีที่รายการข้อมูลมีการ update บ่อยๆ นักศึกษาคิดว่าการ search แบบใดทำงานได้ไวกว่ากัน ให้นักศึกษาลองให้เหตุผลสนับสนุนคำตอบของตนเอง และกำหนดวิธีการทดสอบ