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

ชื่อ – นามสกุล .....เอกรินทร์ องอาจ.....รหัสนักศึกษา.....64015172.....

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	33829
2	69077
3	16899
4	686
5	99844
6	24057
7	13093
8	83357
9	23675
10	54773
ค่าเฉลี่ย	41929

ค่าเฉลี่ย / จำนวนชุดข้อมูลทั้งหมด = 41929 / 100000 = 0.41929

Successfully Search , Worst Case :

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

ครั้งที่	จำนวนครั้งที่เทียบข้อมูล
1	100000
2	100000
3	100000
4	100000
5	100000

6	100000
7	100000
8	100000
9	100000
10	100000
ค่าเฉลี่ย	100000

ค่าเฉลี่ย / จำนวนชุดข้อมูลทั้งหมด = 100000 / 100000 = 1

Unsuccessfully Search :

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

ครั้งที่	**จำนวนครั้งที่เทียบข้อมูล**
1	100000
2	100000
3	100000
4	100000
5	100000
6	100000
7	100000
8	100000
9	100000
10	100000
ค่าเฉลี่ย	100000

ค่าเฉลี่ย / จำนวนชุดข้อมูลทั้งหมด = 100000 / 100000 = 1

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	16
2	17
3	16
4	17
5	16
6	17
7	16
8	17
9	16
10	17
ค่าเฉลี่ย	16.5

ค่าเฉลี่ย / จำนวนชุดข้อมูลทั้งหมด = 16.5 / 100000 = 0.000165

Successfully Search , Worst Case :

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

ครั้งที่	**จำนวนครั้งที่เทียบข้อมูล**
1	16
2	16
3	16
4	16
5	16
6	16
7	16
8	16
9	16
10	16

ค่าเฉลี่ย	16

ค่าเฉลี่ย / จำนวนชุดข้อมูลทั้งหมด = 16 / 100000 = 0.00016

Unsuccessfully Search :

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

ครั้งที่	จำนวนครั้งที่เทียบข้อมูล
1	16
2	16
3	16
4	16
5	16
6	16
7	16
8	16
9	16
10	16
ค่าเฉลี่ย	16

ค่าเฉลี่ย / จำนวนชุดข้อมูลทั้งหมด = 16 / 100000 = 0.00016

ชื่อ – นามสกุล .....เอกรินทร์ องอาจ.....รหัสนักศึกษา.....64015172.....

ตอนที่ 3 : ตอบคำถามและวิเคราะห์การทำงาน

1. Binary search มีข้อจำกัดอย่างไรบ้าง

1.ไม่เหมาะกับการเปลี่ยนแปลงข้อมูลบ่อยๆ
2.ถ้ามี Data ที่น้อย การหาไม่ค่อยต่างกับ o(n)

2. ถ้าเป็นข้อมูลที่ไม่มีการเรียงลำดับ ใน binary search นักศึกษาคิดว่าจะเกิดผลการทำงานเป็นอย่างไร

ถ้าไม่เรียงลำดับมันจะหาข้อมูล แล้วหาไม่เจอเพราะมันอ้างอิงว่าที่เจอมีค่าน้อย หรือมากกว่ามันถ้าไม่เรียงเลยเพี้ยน

3. จากจำนวนข้อมูลที่เท่ากัน ใน worst case และกรณีที่ค้นหาไม่เจอ การทำงานของ search แบบไหนไวกว่ากัน

แบบ linear ยังไงก็ช้ากว่าอยู่ดี เพราะไม่มีการหั่นข้อมูลแบบ binary
Linear เป็นแบบ o(n) ส่วน binary เป็น o(n)

3. จากจำนวนข้อมูลที่เท่ากัน ในกรณีทั่วไป การทำงานของ search แบบไหนไวกว่ากัน

ถ้าจำนวนที่ไม่มาก แบบ linear จะไวกว่าเพราะขั้นตอนการหามันน้อย

4. ในกรณีที่รายการข้อมูลมีการ update บ่อยๆ นักศึกษาคิดว่าการ search แบบใดทำงานได้ไวกว่ากัน ให้นักศึกษาลองให้เหตุผลสนับสนุนคำตอบของตนเอง และกำหนดวิธีการทดสอบ

ในกรณีที่ update ข้อมูลบ่อยๆ แบบ linear จะไวว่าเพราะเอาข้อมูลไปต่อท้ายเสมอไม่สนใจว่าต้องเรียงลำดับหรือไม่ Big O ก็ยังเป็น o(n) เสมอ แต่ต่างจาก binary ที่ต้องการเรียงใหม่ ทุกรอบก่อนจำทำการ search มันทำให้ช้าตรงที่ sort ก่อนเสมอ จากที่ binary เป็นแบบ o(log n) ถ้าเพิ่ม ขบวนการ sort ที่เป็นแบบ quick sort หรือ merge sort ที่เป็น 0(n log n) ที่ทำแบบ linear ที่เป็นแบบ o(n) อีก มันเลยช้ากว่ากัน เพราะฉะนั้นต้องเลือกให้เหมาะกับงานที่สุด

ชื่อ – นามสกุล .....เอกรินทร์ องอาจ.....รหัสนักศึกษา.....64015172.....