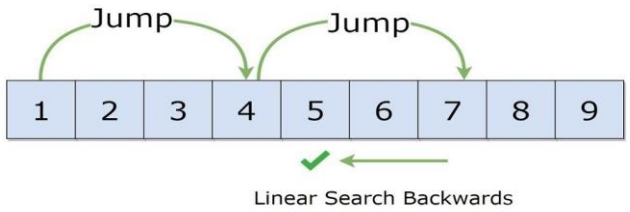
Jump Search



การค้นหาแบบกระโดด

ขั้นตอนวิธี (Algorithm)

- 1.Array ที่นำมา search ต้องได้รับการ sorting(เรียงข้อมูล) แล้ว
- 2.กำหนดตัวแปร size = len(Array)
- 3.$\text{jump\_size} = \sqrt{len(array)}$ กำหนด ตัวแปร jump_size เก็บค่า รากของ size
- 4.นำjump_size มากำหนดเป็น idx มาเข้าเป็นกระบวนการลูปแล้วเพิ่มค่า jump_size ไปเรื่อย เพื่อหา ตัวเป้าหมายแบบกระโดด
- 5. ถ้า ตัวเป้าหมายไม่อยู่ใน Array return -1 ออกมา

```python
def jumpSearch(arr, key):
    size = len(arr)  # todo วัดความยาวของ array
    jump_size = round(size**1/2)  # todo หาขั้นในแต่ละครั้ง
    last_step = 0  # todo เก็บค่าของ jump_size ในกรณีที่ arr[jumpsize] > key
    # todo loop ยังดำเนินเมื่อarr[หาค่าน้อยที่สุด ระหว่าง last_step-1 กับ size] < key
    while arr[min(last_step-1, size)] < key:
        last_step += jump_size  # todo เพิ่ม last_step ด้วย jump_size
        # todo ถ้า last_step >= ขนาดของ arr (ในกรณีที่keyไม่ได้อยู่ใน arr) ส่งค่ากลับเป็น -1
        if last_step >= size:
            return -1
    print('jump size is ',jump_size)
    print('start check ',last_step)
    # todo หา key โดยที่ นำค่า last_step(ที่น้อยกว่า size) มาเริ่มต้นในการหา จนถึงค่าสุดท้าย
    for i in range(last_step, size):
        if arr[i] == key:
            return i  # todo เทียบเจอแล้ว ส่งหากลับเป็น ตำแหน่งของ key
```

Code

ผลการรัน

```
jump size is  3
start check  0
4  at  2
```

```
PS D:\Github\305214-Da
11  at  -1
PS D:\Github\305214-Da
```

กำหนด Array = [1,2,4,5,7,9]

Element ที่ต้องการ คือ 4

Element ที่ต้องการคือ 11 ซึ่งต้อง return -1 ออกมาเพราะไม่อยู่ใน Array

อ้างอิง

https://www.geeksforgeeks.org/jump-search/