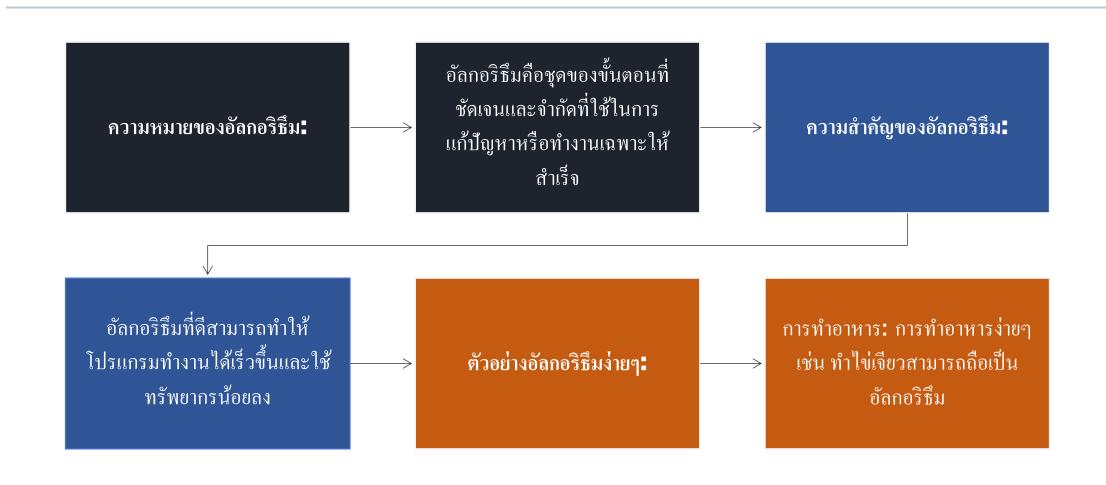


แนะนำอัลกอริธิม

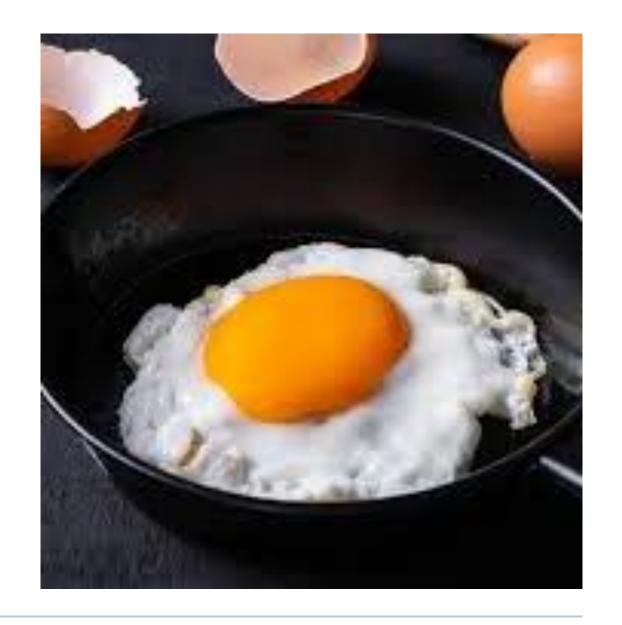


โครงสร้างพื้นฐานของอัลกอริธิม

1. ขั้นตอนการทำงาน:

Algorithm Steps: แบ่งเป็นขั้นตอนที่ ชัดเจน เช่น:

- เตรียมวัตถุดิบ
- ตั้งกระทะ
- ทอดไข่
- เสร็จสิ้น



โครงสร้างพื้นฐานของอัลกอริธิม

2. การใช้คำสั่งเชิงตรรกะ:

• ตัวอย่างโค้ด: การตรวจสอบเงื่อนใจ

```
temperature = 30
if temperature > 25:
print("It's hot outside!")
else:
print("The weather is nice.")
```

คำอธิบาย: ใช้คำสั่ง if เพื่อตัดสินใจว่าความร้อนสูง

โครงสร้างพื้นฐานของอัลกอริธิม

3. การใช้ลูป:

ตัวอย่างโค้ด: การทำงานซ้ำ:

```
for i in range(5):
    print("Iteration", i)
```

คำอธิบาย: ใช้คำสั่ง if เพื่อตัดสินใจว่าความร้อนสูง

การวิเคราะห์ความซับซ้อน

• Time Complexity:หมายถึง: จำนวน ขั้นตอนที่ต้องใช้ในการทำงานของอัลกอริธีม โดยการ วัดในเชิงปริมาณ

คำอธิบาย: Bubble Sort มี Time Complexity เป็น O(n²) เนื่องจากต้องทำการ เปรียบเทียบคู่ค่าทั้งหมด

การวิเคราะห์ความซับซ้อน

- Space Complexity:หมายถึง: จำนวนหน่วยความจำที่ต้องใช้โดยอัลกอริธิม
- ตัวอย่าง: การใช้ตัวแปรในฟังก์ชันข้างต้น ถ้ามีการประกาศตัวแปรมากแสดงถึงการใช้หน่วยความจำที่มากกว่า

ตัวอย่างอัลกอริธีมง่าย

- อัลกอริธิมการหาผลรวมของตัวเลข:
- คำอธิบาย: ใช้ลูป for เพื่อหาผลรวมของตัวเลขจาก 1 ถึง n

```
def sum_numbers(n):
    total = 0
    for i in range(1, n + 1):
        total += i
    return total
```

อัลกอริธีมการคัดกรองเลขคู่:

• คำอธิบาย: ใช้ลูป for และคำสั่ง if เพื่อกรองเลขคู่จากรายการ

```
def filter_even_numbers(numbers):
    even_numbers = []
    for num in numbers:
        if num % 2 == 0:
            even_numbers.append(num)
    return even_numbers
```

การประยุกต์ใช้ในชีวิต จริง

การทำงานประจำวัน:

การทำกิจกรรม
ประจำวัน เช่น การ
จัดการเวลา การวางแผน
กิจกรรม ก็ใช้แนวคิด
ของอัลกอริธึม

การจัดการข้อมูลใน ฐานข้อมูล**:** การค้นหาข้อมูล การ จัดเรียงข้อมูล