Bubble sort

ตัวอย่างการทำงาน

สมมติว่าเราเรามีลิสต์ [5, 1, 4, 2, 8]

รอบที่ 1:

เปรียบเทียบ 5 กับ 1: เนื่องจาก 5 > 1 ให้สลับตำแหน่งกัน จะได้ [1, 5, 4, 2, 8]

เปรียบเทียบ 5 กับ 4: เนื่องจาก 5 > 4 ให้สลับตำแหน่งกัน จะได้ [1, 4, 5, 2, 8]

เปรียบเทียบ 5 กับ 2: เนื่องจาก 5 > 2 ให้สลับตำแหน่งกัน จะได้ [1, 4, 2, 5, 8]

เปรียบเทียบ 5 กับ 8: เนื่องจาก 5 < 8 ไม่ต้องสลับตำแหน่ง จะได้ [1, 4, 2, 5, 8] เหมือนเดิม

สิ้นสุดรอบที่ 1 ตัวเลขที่มากที่สุดคือ 8 จะถูก "bubble" ไปอยู่ที่ตำแหน่งสุดท้ายของรายการแล้ว

รอบที่ 2:

เปรียบเทียบ 1 กับ 4: 1 < 4 ไม่สลับ

เปรียบเทียบ 4 กับ 2: 4 > 2 สลับ จะได้ [1, 2, 4, 5, 8]

เปรียบเทียบ 4 กับ 5: 4 < 5 ไม่สลับ

เปรียบเทียบ 5 กับ 8: 5 < 8 ไม่สลับ (ไม่จำเป็นต้องเปรียบเทียบกับ 8 แล้ว เพราะ 8 อยู่ในตำแหน่งที่ถูกต้องแล้ว)

สิ้นสุดรอบที่ 2 ตัวเลข 5 จะถูก "bubble" ไปอยู่ในตำแหน่งที่ถูกต้อง

รอบที่ 3:

เปรียบเทียบ 1 กับ 2: 1 < 2 ไม่สลับ

เปรียบเทียบ 2 กับ 4: 2 < 4 ไม่สลับ

เปรียบเทียบ 4 กับ 5: 4 < 5 ไม่สลับ

สิ้นสุดรอบที่ 3 จะไม่มีการสลับเกิดขึ้นเลย ซึ่งเป็นสัญญาณบ่งบอกว่ารายการถูกเรียงลำดับเสร็จสมบูรณ์แล้ว แต่โปรแกรมยังทำงานต่อไป ด้วย **for i in range(n):**

สมมติว่าเรามีลิสต์ [64, 34, 25, 12, 22, 11, 90]

รอบที่ 1:

(64, 34) -> สลับ -> [34, 64, 25, 12, 22, 11, 90]

(64, 25) -> สลับ -> [34, 25, 64, 12, 22, 11, 90]

(64, 12) -> สลับ -> [34, 25, 12, 64, 22, 11, 90]

(64, 22) -> สลับ -> [34, 25, 12, 22, 64, 11, 90]

(64, 11) -> สลับ -> [34, 25, 12, 22, 11, 64, 90]

(64, 90) -> ไม่สลับ -> [34, 25, 12, 22, 11, 64, 90]

หลังรอบที่ 1: 90 (ตัวที่ใหญ่ที่สุด) จะอยู่ที่ตำแหน่งสุดท้าย

รอบที่ 2: (พิจารณาถึงตำแหน่งก่อน 90)

(34, 25) -> สลับ -> [25, 34, 12, 22, 11, 64, 90]

(34, 12) -> สลับ -> [25, 12, 34, 22, 11, 64, 90]

(34, 22) -> สลับ -> [25, 12, 22, 34, 11, 64, 90]

(34, 11) -> สลับ -> [25, 12, 22, 11, 34, 64, 90]

(34, 64) -> ไม่สลับ -> [25, 12, 22, 11, 34, 64, 90]

หลังรอบที่ 2: 64 จะอยู่ที่ตำแหน่งรองสุดท้าย

กระบวนการนี้จะดำเนินต่อไป ด้วย **for i in range(n):**