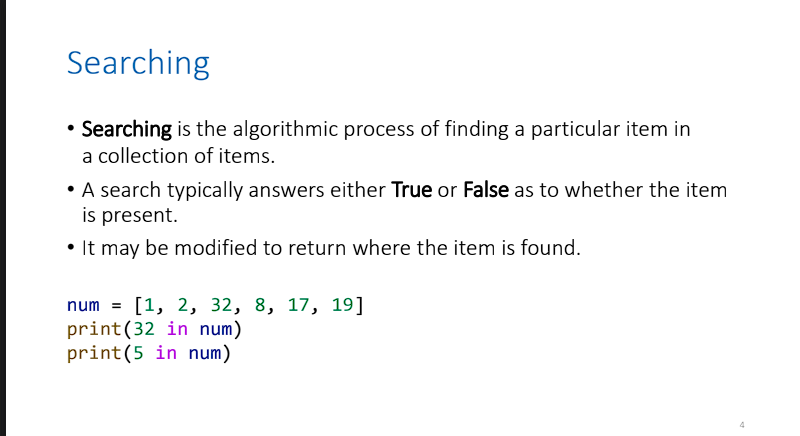
การค้นหาข้อมูล

เช็คคนที่มีคะแนน 15 คะแนน

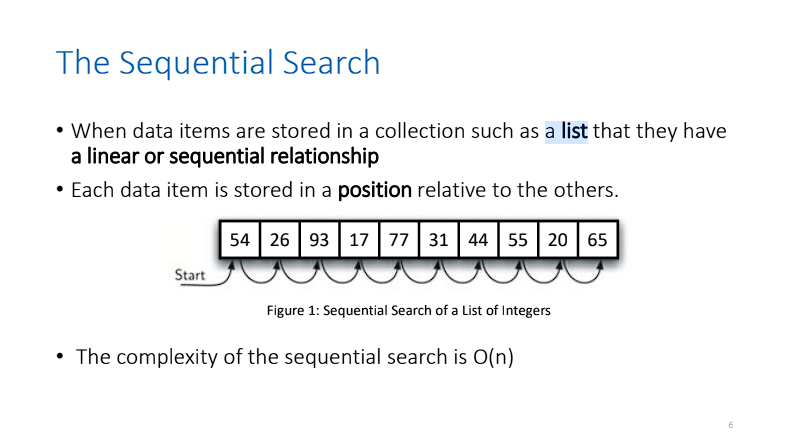
การคืนค้าข้อมูล แบบ true false และ เก็บเป็นข้อมูล

Sequential search



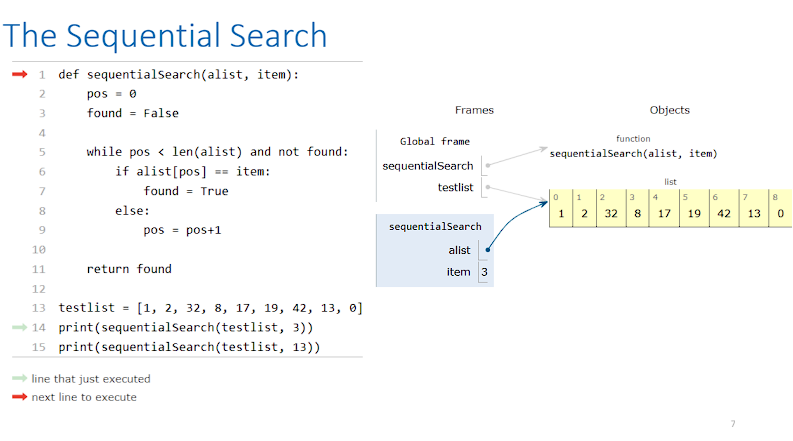
Num = list

ใช้คำสั่ง in ในการค้นหาข้อมูล เช่น หา 32 in Num



Big o = o(n)

หาข้อมูลแบบเรียงลำดับ



13 หาค่า 3 ไร้หาตั้งแต่ตัวแรกไปยันตันสุดท้าย

Alist เก็บ testlist

Item เก็บ 3

Pos = positions

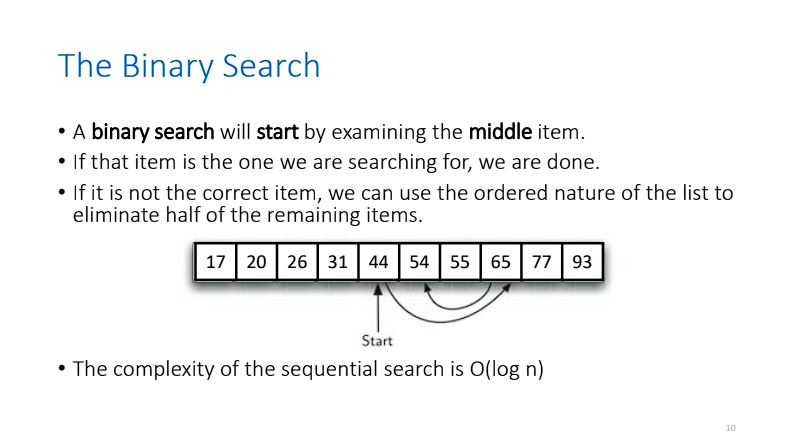
Found = true คือ หาตัวเลขที่ต้องการเจอแล้ว

Pos = pos + 1 เพิ่มตำแหน่งถัดไปเพื่อหาข้อมูล

3 while loop 10 รอบ

13 while loop 9 รอบ

Binary search (log n)



เป็นการแบ่งครึ่งหาข้อมูล, หาจากค่า middle, ซอยแบ่งครึ่งหาข้อมูลไปเรื่อย ๆ

index 0 = 17 –index 9 = 93

สูตร (first + last) // 2

1. index 0 \* 9 / 2 = 4 = 44

หากข้อมูลที่หามากกว่าตรงกลาง จะถูกตัดทิ้งด้วย

2. index 5 + 9 / 2 = 7 = 65

3. last = middle – 1 = index = 5 = 54

4. first = last = index 6 = 55

Check in while loop = 4 ครั้ง

Check out while loop = 5 ครั้ง

1. pos 4 = 13 (0+8 // 2 = 4)

2. pos 1 = 1 (0+3 // 1 = 1)

3. pos 2 = 2 (1+3 // 2 = 2)

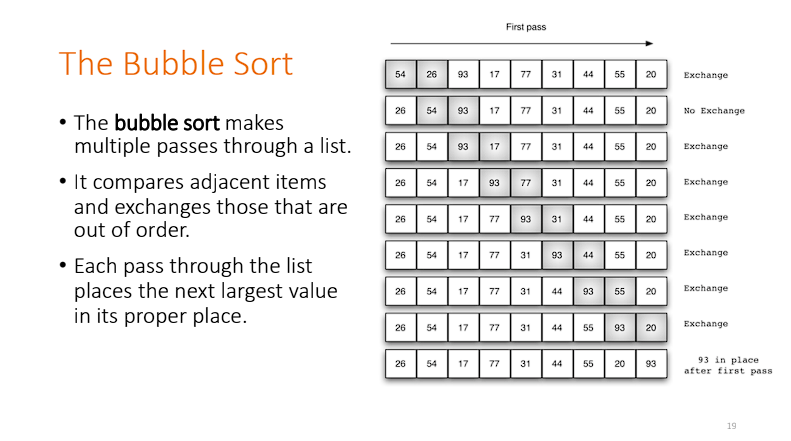
4. pos 3 = 8 (3+(-3) // 3 = 8)

1. pos 4 = 1 = 17 – 93 = 44

2. pos 1 = 20 = 17 – 31 = 20

**Sorting**

การเรียงข้อมูล เรียงตามตัวอักษร เรียงตามตัวเลข

**Bubble sort **

การเปรียบเทียบที่ละคู่

เรียกว่า การ swap ค่า โดย หากเลขทางด้ายซ้ายมากกว่าจะสลับที่กัน แต่ถ้าเลขทางด้านซ้ายน้อยกว่า จะเปลี่บนตัวเลขที่ใช้เปรียบเทียบ

26 17 54 31 44 55 20 77 93

17 26 31 44 54 20 55 77 93

17 26 31 44 20 54 55 77 93

17 26 31 20 44 54 55 77 93

17 26 20 31 44 54 55 77 93

17 20 26 31 44 54 55 77 93

54 26

I = 54

J = 26

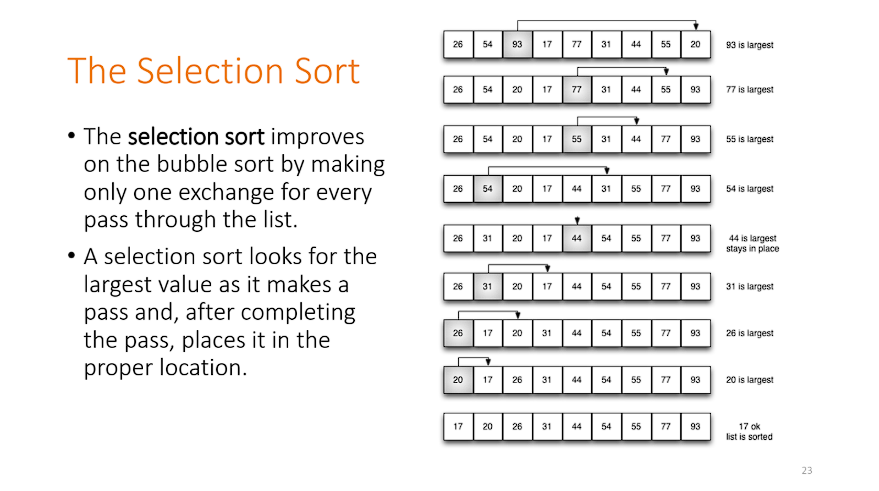
Temp = I 54

i = j 54 26

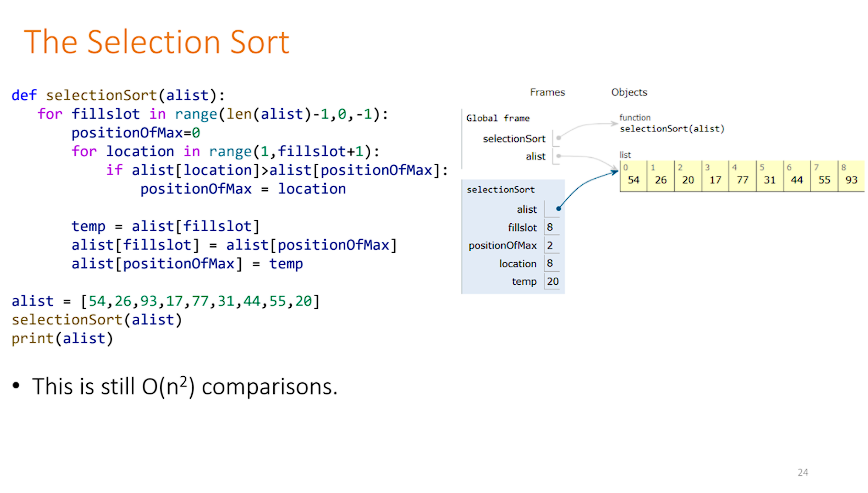
j = temp

26 54

**Selection sort**



ไม่ swap ทุกครั้งที่เปรียบเทียบค่าตัวเลข แต่จะเรียงค่าทันที เช่น น้อยไปมาก จะย้ายค่ามากที่สุด(93)ไปตำแหน่งสุดท้าย(20)และทำการแลกที่กันระหว่าง 93 กับ 20 และลดขอบเขตในการพิจารณา เช่น รอบแรก พิจารณา 9 รอบ รอบที่ สองพิจารณา 8 รอบ



position max เปรียบเทียบตัวเลขที่มีค่าที่สุดก่อน จากนั้นค่อยทำการ swap ค่าภายหลัง

Fillslot เก็บค่าตำแหน่งสุดท้าย