

สมาชิกในกลุ่ม

- 66070228 อินทิรา ธนพประภัศร์
- 66070248 ชนิสรา จันทรคำจร
- 66070286 ปณิสยา บุญประกอบ

Miau

Code block

1. Selection Sort

```
@main def main(): Unit =  
  println(selectionSort(List(5, 3, 8, 6, 2)))  
  println(selectionSort(List(9, 7, 6, 4, 1, 2, 3)))  
  
def selectionSort(xs: List[Int]): List[Int] = xs  
  match {  
    case Nil => Nil  
    case _ =>  
      val min = xs.min  
      min :: selectionSort(xs.filterNot(_ == min))  
  }
```

2. Insertion Sort

```
def insertion_sort(list: List[Int]): List[Int] = {  
  def insert(x: Int, sorted: List[Int]): List[Int] = sorted  
    match  
    {  
      case Nil => List(x)  
      case head :: tail =>  
        if (x < head) x :: sorted  
        else head :: insert(x, tail)  
    }  
  
  list match  
  {  
    case Nil => Nil  
    case head :: tail => insert(head, insertion_sort(tail))  
  }  
}
```

```
UnsortList : List(5, 3, 1, 2, 4)  
List(1, 2, 3, 4, 5)  
UnsortList : List(-5, 10, -10, 8, 1)  
List(-10, -5, 1, 8, 10)
```

Output ที่ได้

```
List(2, 3, 5, 6, 8)  
List(1, 2, 3, 4, 6, 7, 9)
```

1. ฟังก์ชัน selectionSort รับค่าเป็น list โดยให้ตัวแปรชื่อ xs
2. ใช้ match เพื่อตั้ง case ที่จะเกิดขึ้น
 - a. ถ้าเป็น Nil (หรือ Null) จะ return ค่า Nil ออกมา
 - b. ถ้าไม่เป็น Nil ให้ใช้ฟังก์ชัน min เพื่อคืนค่า min ทุกครั้งที่เรียก recursion จากนั้นเอา min ไปต่อข้างหน้า list ที่ถูก sort แล้ว จนกว่า list เดิมจะเป็น Nil

ฟังก์ชัน insertion_sort รับค่าเป็น list

- ถ้าเป็นลิสต์ว่าง จะส่งคืนค่าลิสต์ว่าง
- ถ้าไม่ จะแบ่งเป็น head กับ tail แล้วเรียกใช้ฟังก์ชัน insert โดยเป็น insert(head, insertion_sort(tail))

ฟังก์ชัน insert รับค่า x เป็น Int, รับค่า sorted เป็น list แทนลิสต์ที่เรียงแล้ว

- ถ้า sorted เป็นลิสต์ว่าง จะส่งคืนค่า List(x)
- ถ้าไม่ จะแบ่ง head กับ tail แล้วทำการเปรียบเทียบ
 - ถ้า x น้อยกว่า head ให้แทรก x ไปไว้ด้านหน้า
 - ถ้า x มากกว่าหรือเท่ากับ head ให้ head อยู่ที่เดิม แล้วเรียกใช้ insert(x, tail) เพื่อแทรกค่า x โดยเปรียบเทียบกับค่าที่เหลือ ซึ่งค่าที่เหลือก็จะเป็น head เพื่อทำการเปรียบเทียบ