

ใบงานที่ Transform

วัตถุประสงค์ เพื่อศึกษาการสังเคราะห์ข้อมูล และการแก้ปัญหาด้วยกลยุทธ์ greedy

ปัญหา knapsack มี
จำนวนวิธีหิยได้ 2^n
(วิธีการแจกแจงตาม
หลักการ brute force)

เราสามารถลดปริมาณ
งานด้วยการเรียงมูลค่า
ต่อน้ำหนัก (หิยของที่
มีมูลค่าต่อน้ำหนัก
สูงสุ่ก่อน)

จากส่วนของโปรแกรม
ประกอบ เราคำนวณ
perW โดยมี isPicked
สร้างมาเพื่อเก็บว่าของ
ชื่ออะไร (label) จะถูก
หิย

ให้ถือว่าต้นทุนการเรียง

(ตัวอย่างนี้เรียงอยู่แล้ว) เป็น $O(n \log n)$ การพิจารณาหิยใส่สุ่ตรงใดที่หิยใส่แล้วน้ำหนักไม่เกิน จะพิจารณา n ครั้ง ดังนั้น $O(n \log n) + O(n) < 2^n$

หมายเหตุ การหิยตามลำดับสัดส่วนมูลค่าต่อน้ำหนักถือเป็นการแก้ปัญหาแบบ greedy

ตอบ Q1

$\text{sumW} + \text{weights}[k] \leq C$

ตอบ Q2

$\text{sumV} += \text{values}[k];$
 $\text{sumW} += \text{weights}[k];$
 $\text{isPicked}[k] = \text{true};$

```
char[] labels = { 'a', 'b', 'c', 'd' };
int[] values = { 20, 30, 50, 10 };
int[] weights = { 2, 5, 10, 5 };
double[] perW = new double[values.length];
boolean[] isPicked = new boolean[values.length];
int C = 16;
int sumV = 0;
int sumW = 0;

//default value of boolean array is "false"
System.out.println(Arrays.toString(isPicked));
for (int i = 0; i < values.length; i++)
    perW[i] = (double) values[i] / weights[i];
int k = 0;
while ( /* Q1 */ ) {
    /* Q2 */
    k++;
}
System.out.println("sum value is "
    + sumV + " sum weight is " + sumW);
```

ตอบ Q3 คำตอบที่ได้ เหมือนหรือต่างจากคำนวณด้วยการ Brute Force

แตกต่างกับการ Brute Force เพราะ greedy เลือกตามมูลค่าต่อน้ำหนักไม่ใช่คำตอบที่ดีที่สุ่แต่ใช้เวลาน้อยกว่า Brute Force Brute Force แลagmaด้วยเวลาเพราะขึ้นอยู่กับจำนวน input แต่รับประกันคำตอบที่ดีที่สุ่