

ใบงานที่ Transform

วัตถุประสงค์ เพื่อศึกษาการสังเคราะห์ข้อมูล และการแก้ปัญหาด้วยกลยุทธ์ greedy

ปัญหา knapsack มี
จำนวนวิธีหิยได้ 2"
(วิธีการแจกแจงตาม
หลักการ brute force)

เราสามารถลดปริมาณ
งานด้วยการเรียงมูลค่า
ต่อน้ำหนัก (หิยของที่
มีมูลค่าต่อน้ำหนัก
สูงสุดก่อน)

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

ให้ถือว่าต้นทุนการเรียง

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

หมายเหตุ การหิยตามลำดับสัดส่วนมูลค่าต่อน้ำหนักถือเป็นการแก้ปัญหาแบบ greedy

ตอบ Q1 $\text{sumW} < C \ \&\& \ ((\text{sumW} + \text{weights}[k]) < C)$

ตอบ Q2

```
sumW += weights[k];
sumV += val[k];
```

ตอบ Q3 คำตอบที่ได้ เหมือนหรือต่างจากคำนวณด้วยการ Brute Force

ต่างกัน

```
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);
```