练习3 – 0-1背包

- 有 n 件物品和一个最大承重为 W 的背包，每件物品的重量是 w_i、价值是 v_i
  - 在保证总重量不超过 W 的前提下，将哪几件物品装入背包，可以使得背包的总价值最大?
  - 注意：每个物品只有 1 件，也就是每个物品只能选择 0 件或者 1 件，因此称为 0-1背包问题

- 如果采取贪心策略，有3个方案

① 价值主导：优先选择价值最高的物品放进背包

② 重量主导：优先选择重量最轻的物品放进背包

③ 价值密度主导：优先选择价值密度最高的物品放进背包（价值密度 = 价值 ÷ 重量）

0-1背包 – 实例

■ 假设背包最大承重150，7个物品如表格所示

编号	1	2	3	4	5	6	7
重量	35	30	60	50	40	10	25
价值	10	40	30	50	35	40	30
价值密度	0.29	1.33	0.5	1.0	0.88	4.0	1.2

① 价值主导：放入背包的物品编号是 4、2、6、5，总重量 130，总价值 165

② 重量主导：放入背包的物品编号是 6、7、2、1、5，总重量 140，总价值 155

③ 价值密度主导：放入背包的物品编号是 6、2、7、4、1，总重量 150，总价值 170

小码哥教育 SEEMYGO

0-1背包 – 实现

```
void run(String titile, Comparator<Article> comparator) {
    Article[] articles = new Article[] {
        new Article(35, 10), new Article(30, 40),
        new Article(60, 30), new Article(50, 50),
        new Article(40, 35), new Article(10, 40),
        new Article(25, 30)
    };
    Arrays.sort(articles, comparator);
    int capacity = 150, weight = 0, value = 0;
    List<Article> selectedArticles = new ArrayList<>();
    for (int i = 0; i < articles.length && weight < capacity; i++) {
        int newWeight = articles[i].weight + weight;
        if (newWeight <= capacity) {
            selectedArticles.add(articles[i]);
            weight = newWeight;
            value += articles[i].value;
        }
    }
    System.out.println("------------" + titile + "------------");
    System.out.println("总价值: " + value);
    for (Article article : selectedArticles) {
        System.out.println(article);
    }
}
```

0-1背包 – 实现

```
run("重量主导", (Article a1, Article a2) -> {
    return a1.weight - a2.weight;
});

run("价值主导", (Article a1, Article a2) -> {
    return a2.value - a1.value;
});

run("价值密度主导", (Article a1, Article a2) -> {
    return Double.compare(a2.valueDensity, a1.valueDensity);
});
```

```
------------重量主导------------
总价值：155
[weight=10, value=40]
[weight=25, value=30]
[weight=30, value=40]
[weight=35, value=10]
[weight=40, value=35]
------------价值主导------------
总价值：165
[weight=50, value=50]
[weight=30, value=40]
[weight=10, value=40]
[weight=40, value=35]
------------价值密度主导------------
总价值：170
[weight=10, value=40]
[weight=30, value=40]
[weight=25, value=30]
[weight=50, value=50]
[weight=35, value=10]
```

小码哥教育 SEEMYGO

0-1背包 – 实现

```java
public class Article {
    int weight;
    int value;
    double valueDensity;
    public Article(int weight, int value) {
        this.weight = weight;
        this.value = value;
        valueDensity = value * 1.0 / weight;
    }
    @Override
    public String toString() {
        return "[weight=" + weight + ", value=" + value + "]";
    }
}
```