Problem 1.

$V = \{e_1, \cdots, e_n\}$ a universe of elements.

$\{S_i \subseteq V\}_{i=1}^{m}$: a list of (possibly overlapping) sets

Decision Variables:

x_i : a binary decision variable of whether to include the set i.

Objective : cover all elements with minimum number of sets.

$$\min \sum_{i=1}^{n} x_i$$

$$\text{s.t.} \quad \bigcup_{i=1}^{n} x_i S_i = V$$

```
Restricted license - for non-production use only - expires 2022-01-13
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 8 rows, 5 columns and 13 nonzeros
Model fingerprint: 0x74c2bfc5
Variable types: 0 continuous, 5 integer (5 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 4.0000000
Presolve removed 8 rows and 5 columns
Presolve time: 0.00s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.00 seconds
Thread count was 1 (of 8 available processors)

Solution count 1: 4

Optimal solution found (tolerance 1.00e-04)
Best objective 4.000000000000e+00, best bound 4.000000000000e+00, gap 0.0000%

    Variable            X
-------------------------
if_chooseSet[1]           1
if_chooseSet[2]           1
if_chooseSet[3]           1
if_chooseSet[4]           1
PS C:\Users\71401\Desktop>
```

Problem 2.

Suppose:

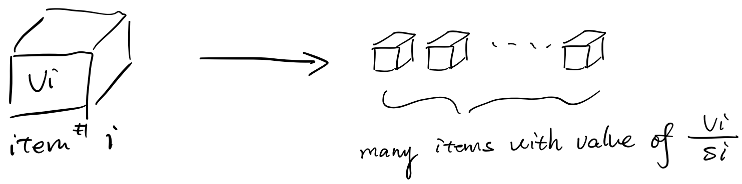
- n items $\{1, \cdots, n\}$
- Each item i has a value V_i and size S_i
- A capacity B (for sizes)

Goal: Find a max-value subset S from n items.

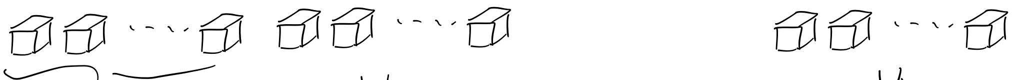
Greedy algorithm tells us to keep choosing items with highest density.

We need to show that greedy algorithm is optimal for the Fractional Knapsack Problem.

Since we can take fractional weight, we can devide all the items into unit size items.



After the devision, we sort the unit-size items according to their value.



$$\frac{V_a}{S_a} > \frac{V_b}{S_b} > \cdots > \frac{V_n}{S_n}$$

To get a higher ~~total~~ value, obviously choosing unit items with higher value will leads to optimal solution.

This is exactly what we are doing in greedy method.

Therefore, we can say that greedy rule is optimal for the Fractional Knapsack Problem.