問題4

Fractional knapsack problem

Exercises 15.2-6

- Show how to solve the fractional knapsack problem in O(n) time.

Fractional knapsack problem

- 給定n個不同物品，每個物品都有其重量(weight)、價值(value)。
- 現有一背包，其最大承重量W。
- 問題: 要將哪些物品放入背包才能讓背包不超重又能有最大價值?
- (物品可以只取其一部分)

Example

Items	a	b	c	d	e
Value	4	2	7	8	3
Weight	1	2	3	4	5

W(capacity): 8

Solution

- 將每個物品的價值(v)除以其權重(w)
- 依序挑選值權(v/w)比最高的物品直到背包裝不下

Solution

Items	**a**	**b**	**c**	**d**
Value	4	2	7	8
Weight	1	2	3	5
v/w	4	1	2.3	1.6

1. Take item **a**, space left: 7
2. Take item **c**, space left: 4
3. Take 4/5 of item d , space left: 0
4. Item taken: a, c, 4/5d. Total value: 17.4

Problem

- The time taking step is the sorting of all items in decreasing order of their v/w ratio.
- The average time complexity of QuickSort is O(nlogn).

QuickSelect

- Based on QuickSort, but doesn’t need to sort the entire array.
- Time complexity: O(n)

Time complexity

- Find the median item from {v/ws} (O(n))
- See if you can fill the knapsack with items that are more valuable than the median (O(n))
- If you can, do so and recursively solve the problem for the n/2 items of lower value given that you’ve already filled the knapsack.
- If you can’t, then you can throw out the n/2 items of lower value, and then try to solve the problem again with only the n/2 items of higher value.
- $T(n) = T(n/2) + O(n) = O(n)$

Implementation

- Chose the median **r** from **R** (set of v/w ratios)
- Determine
    - R1 = {v/ws that are larger than **r**}, W1 = sum of their weights
    - R2 = {v/ws that are equal to **r**}, W2 = sum of their weights
    - R3 = {v/ws that are smaller than **r**}, W3 = sum of their weights
- If W1 > W
    - Recurse on R1
- Else
    - While (knapsack isn't full and R2 is not empty)
        - Add items from R2
    - If (knapsack gets full)
        - Return items in R1 and those just added from R2
    - Else
        - Reduce W by W1+W2
        - Recurse on R3 and return items in R1 and R2
        - Add items returned from recursive call