Greedy Approach

- Constraint
- Optimization Problem (Max, Min)

1) Fractional Knapsack

constraint: total Weight <= M

Max Profit

Complexity – $n \log n$

Approach

① for(int i = 0 to n-1){ — n
P/W (Ratio)
}

② Sort P/W Dec order — $n \log n$

③ for (i = 0 to n-1){ M decrease – till the value becomes 0 — n
totalProfit increase
}

Weight
10

M = 9

9/10 – fractional Knapsack

Job Scheduling

Job — Deadline

Job — Profit — Max Profit

min(1, 6) = 1

Job	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9
Profit	55	65	75	60	70	50	85	68	45
Deadline	5	2	7	3	2	1	4	5	3

1	2	3	4	5	6	7
J_2	J_5		J_7	J_8		J_3

T ~~f~~ ~~f~~ f ~~f~~ f f f

0 1 T 2 T 3 4 5 0

Job Id →	1	2	3	4	5	6	7	8	9
Deadline →	5	2	7	3	2	1	4	5	3
Profit →	55	65	75	60	70	50	85	68	45

Constraint → Need to complete a job within given deadline.

Optimization → Max Profit

1) Sort the jobs → Profit → Decreasing order

Job Id	7	3	5	8	2	4	1	6	9
Deadline	4	7	2	5	2	3	5	1	3
Profit	85	75	70	68	65	60	55	50	45

min(3,6)

min(6,6) = 6

min(1,6) = 1

min(1,6) = 1

min(4,6) = 4

min(0,6) = 0

max Deadline = 7

2)

Ids	0	1	2	3	4	5	6
Job →	2	5	4	7	8	–	3
result →	~~f~~ T	~~f~~ T	~~f~~ T	~~f~~ T	~~f~~ T	f	~~f~~ T

min(maxDeadline - 1, arr(i).get.deadline(i) - 1)

Merge Intervals

start end end end end

(1, 3) (2, 6) (8, 10) (15, 18)

start start start

end > start — overlapping

end < start — Non-overlapping

{ 7 - 9, 8 - 10 } ⇒ 7 - 10

(1, 4) (2, 3) — interval

true

(1, 4) $4 > 2$

interval

(1, —

(8, 10) (15, 18)

$10 < 15$

```
for(int[] interval: intervals){
        // No overlapping
        // 10 < 15 -yes
        if(merged.isEmpty() || merged.getLast()[1] < interval[0]){
           merged.add(interval);
        }
        else{
           // overlapping
           // max(LastEnd, end)
           // [1, 6]
           // [1, 4] [2, 3] - [1, 4]
           merged.getLast()[1] = Math.max(merged.getLast()[1], interval[1]);
        }
```

true

Maximum Value return

Huffman coding

Huffman tree

Data compression

Max frequency — Lower bit

Low frequency — Higher bit

Optimal Merge Pattern

Optimization

$f1 = 5$

$F2 = 7$

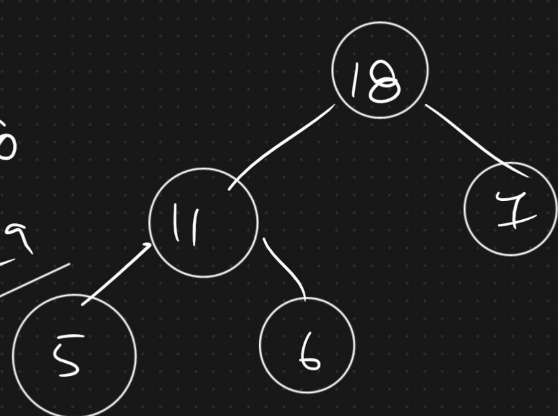
$F3 = 6$

Minheap

Priority Queue

$11+18$

$= 29$



Assignment Problem

https://leetcode.com/problems/task-scheduler/description/

Quicksort Algorithm

Live session