Algorithmic Paradigm: Greedy Algorithms

1. Pruning — Linear Search, Binary search, 3-way

$n \to n-1$, $n \to n-a$, $n \to n/2$, $n \to n/3$

2. Incremental Design — Incremental Sort: Insertion Sort, Selection Sort, Bubble Sort

3. Divide and Conquer — Merge Sort, Quick Sort

4. Greedy Algorithms

5. Dynamic Programming

Algorithmic Paradigm :- Greedy Algorithms

↳ optimization Problems

Minimization / Maximization

↳ Decision Problems

Decision :- I/P Array, x
? $x \in A$
O/P: Yes / No

I/P: Integer p
? Is p a prime
O/P: Yes / No

Optimization Problems :- Minimization → | Maximization

I/P: Road N/W
? Find Min Cost tour from A to B.
O/P:- Shortest Path (A, B).

P

Algorithmic Paradigm:- Greedy Algorithms

③

↳ Greedy Strategy / Approach,
(Heuristics)
+ Proof of Correctness

} Algorithm

Greedy Algorithm:- Global optimum
↳ Min/Max

$$\underbrace{\text{local opt}_1 \rightarrow \text{local opt}_2 \rightarrow \cdots \rightarrow \text{Global optimum}}_{\text{Proof of Correctness.}}$$

Algorithmic Paradigm:- Greedy Algorithms

Coin change Problem:- I/P: Integer x, $d_1, d_2, \ldots d_k$

? Change for x

objective:- Minimize no. of coins used

$x = 100 \qquad d_1 = 5,\ d_2 = 2,\ d_3 = 1$

Greedy strategy:- Supply Max no. of higher denomination coins

$\left\lfloor \frac{100}{5} \right\rfloor = 20$

$\text{change}(100) = (20, 0, 0)$

$x = 117 \quad \left\lfloor \frac{117}{5} \right\rfloor = 23, \rightarrow y = 117 - 23 \times 5 = 2, \rightarrow \frac{y}{2} = 1$

$\text{change}(117) = (23, 1, 0)$

P

Algorithmic Paradigm:- Greedy Algorithms

$$\begin{array}{l} \quad\quad\quad\quad\quad\quad\; 5 \quad 2 \quad 1 \\ \text{Change}(29) = (5, 2, 0) \end{array}$$

Change(36) = (7, 0, 1)

Change(4) = (0, 2, 0)

I/P:- x, (5, 2, 1)

Greedy strategy is optimal

it works always for any $x \geq 1$.

I/P:- x, (5, 4, 2, 1)

$x = 13$ Change(13) = (2, 0, 1, 1) → 4 coins Change(13) = (0, 3, 0, 1) → 4 coins

Message:- We may find more than one opt.

Change(13) = (1, 2, 0, 0) - 3 coins.

Message:- By using Greedy, we may get feasible but not optimal.

Feasible ✓

optimal ✗

Algorithmic Paradigm :- Greedy Algorithms

I/P :- Integer x, Denoms $(5, 4, 1)$

$x = 12$ change$(12) = (2, 0, 2)$ Is this opt :- Change$(12) = (0, 3, 0)$

\# Coins $= 4$ \# Coins $= 3$

$x = 13$ Change$(13) = (2, 0, 3)$ Is this opt Change$(13) = (0, 3, 1)$

\# Coins $= 5$ \# Coins $= 4$

Greedy strategy Fails for $(5, 4, 1)$

works for some $(d_1, d_2, \ldots d_k)$
+ Proof of Correctness
= Greedy Algorithm

doesn't work for some $(d_1', d_2', \ldots d_k')$

→ Bruteforce

→ dynamic Programming.

Knapsack Problem

I/P: $S = \{x_1, x_2, --- x_n\}$ - objects

$w_1, w_2, --- w_n$ - weights

$P_1, P_2, --- P_n$ - Profits

objective :- Find $S' \subseteq S$ s.t.

Profit(S') is Maximum

Constraint :- weight$(S') \leq W$

$S = \{x_1, x_2, x_3\}$

w_i	100	500	400
P_i	1600	1400	8000

$W = 700$

$\boxed{\text{Not Feasible } \{x_2, x_3\}}$

optimal.

$S' = \{x_1, x_3\}$

$w_i = 100 + 400 \leq 700$ ⇒ Maximum Profit.

$P_i = 1600 + 8000 = 9600$

Bruteforce :- Try all Subsets of S

$w(S') \leq W$

Knapsack Problem

Greedy strategy :- Pack as many object as possible.

Sort the weight in Increasing (Non-decreasing) order

$$w_1 \le w_2 \le w_3 \le \cdots \le w_n$$

$S = x_1, x_2, x_3, x_4$

$w = 1, 2, 4, 5$ $\quad W = 6$

$P = 10, 20, 40, 50$

$1 \le 6$ Include x_1

$1+2 \le 6$ Include x_2

$1+2+4 \not\le 6$ Stop

O/P: $S' = \{x_1, x_2\}$

$Profit(S') = 30$ → Is this Maximum

$S' = \{x_1, x_4\}$

Weight $1+5 \le 6$

Profit $10+50 = 60$

Optimal.

∴ Above Greedy strategy Fails

Knapsack Problem

Greedy strategy :- Greedy w.r.t Profit (Pack the highest Profit item First).

Sort the Profits in Decreasing (Non-Increasing) order.

$S = x_4 \quad x_3 \quad x_2 \quad x_1$

$w = 5 \quad 4 \quad 2 \quad 1 \qquad W = 6$

$P = 50 \quad 40 \quad 20 \quad 10$

$5 \leq 6$ Include x_4

$5+4 \nleq 6$ Exclude x_3

$5+2 \nleq 6$ Exclude x_2

$5+1 \leq 6$ Include x_1

$S' = \{x_1, x_4\}$ $10+50 = 60$ optimal.

Does this strategy always work ?

For eg :-

$S = \{x_1, x_2, x_3, x_4\}$

$P = \{10, 25, 40, 50\}$

$W = \{1, 2, 4, 5\}$

$\therefore$ Above Greedy strategy Fails

Knapsack Problem

GS1: Greedy w.r.t. weight

GS2: Greedy w.r.t. Profit

GS3: Greedy w.r.t. $\frac{P_i}{W_i}$

Sort $\frac{P}{W}$ in decreasing (Non-Increasing) order

$$\frac{P_1}{W_1} \geq \frac{P_2}{W_2} \geq \cdots \geq \frac{P_n}{W_n}$$

$S = \{x_1 \quad x_2 \quad x_3 \quad x_4\}$

$W = \{1 \quad 2 \quad 4 \quad 5\}$

$P = \{10 \quad 25 \quad 40 \quad 50\}$

$\frac{P_i}{W_i} = \{10 \quad 12.5 \quad 10 \quad 10\}$

$S' = \{x_2, x_3\}$

$W = 25 + 40 = 65$

$S' = \{x_2, x_1\}$

$W = 25 + 10 = 35$

$S = \{x_1 \quad x_2 \quad x_3 \quad x_4\}$

$W = \{1 \quad 2 \quad 4 \quad 5\}$

$P = \{10 \quad 20 \quad 40 \quad 50\}$

$\frac{P_i}{W_i} = \{10 \quad 10 \quad 10 \quad 10\}$.

$\rightarrow$ GS3 Fails

Knapsack Problem

Variant 1

0/1 - Knapsack problem

each $x_i \in S' - 1$

(or)

$x_i \notin S' - 0$

Variant 2

Fractional Knapsack Problem.

$\alpha\, x_i$

$\hookrightarrow 0 \leq \alpha \leq 1$

$x_1, x_2, 0.5\, x_3$

→ All the three Greedy strategies (GS1, GS2, GS3) fails for 0/1 - Knapsack Problem.

→ G.S 3 (i.e. $\frac{P_i}{w_i}$) works for Fractional Knapsack problem all the time to get an optimal solution.

Fractional Knapsack Problem

GSS3:- $\frac{P_i}{W_i}$ works - Sort $\frac{P_1}{W_1} \geq \frac{P_2}{W_2} \geq \cdots \geq \frac{P_n}{W_n}$

$S = \{x_1, x_2, x_3, x_4\}$

$W = \{1, 2, 4, 5\}$ $\quad W = 6$

$P = \{10, 25, 30, 35\}$

$\frac{P_i}{W_i} = \{10 \quad 12.5 \quad 7.5 \quad 7\}$

W_2 $\quad 2 \leq 6$ $\quad$ Include x_2 $\quad \alpha x_2 : \alpha = 1$

$W_2 + W_1$ $\quad 2+1 \leq 6$ $\quad$ Include x_1 $\quad \alpha x_1 : \alpha = 1$

$W_2 + W_1 + \frac{3}{4} W_3$ $\quad 2+1+\frac{3}{4}\cdot 4 \leq 6$ $\quad$ Include x_3 $\quad \alpha x_3 : \alpha = \frac{3}{4}$

Profit:- $25 + 10 + \frac{3}{4}\cdot 30 = 57.5$ $\quad$ Optimal for Fractinal knapsack.

P