

A decorative graphic on the left side of the slide, consisting of a network of light blue lines and small circles, resembling a circuit board or a stylized tree structure, set against a blue gradient background.

# MORE DP & GREEDY

# DP

Yang dibahas :

- Backtrack Solusi(cara mendapat solusi optimal)
- Optimisasi
  - Flying Table
  - State Reduction
  - DP pakai MAP
  - Optimisasi dengan DS

# BACKTRACK SOLUSI

- Ada 2 cara buat backtrack solusi :
- Simpen jalan optimal rekursi di array lain
- Jalanin lagi rekursi nya tapi Cuma jalan ke state yang optimal

# FLYING TABLE

- DP dengan state lebih dari 1, missal  $(x, y)$
- Untuk cari nilai  $(x, *)$  hanya butuh dari  $(x-1, *)$  atau  $(x, *)$

# FLYING TABLE

Maka dari itu tidak perlu menggunakan memori  $x*y$ , hanya butuh sebanyak range pencarian dari  $x$  ke belakangnya dikali  $y$ .

# FLYING TABLE

Contoh Lain :

mencari LCS dari dua string S dan T

$1 \leq |S|, |T| \leq 10.000$

LCS kan butuh memory  $|S| * |T| = 10^8$

# FLYING TABLE

Tapi untuk nyari  $LCS(x, y)$ , cuma butuh  $LCS(x-1, y)$ ,  $LCS(x, y-1)$ , sama  $LCS(x-1, y-1)$

maka kita bisa mengurangi menjadi  $2*|T|=20000$



# DP PAKAI MAP

biasanya DP mengunjungi semua state

tetapi kadang DP hanya mengunjungi sedikit macam state



# DP PAKAI MAP

CONTOH :

anda punya  $N$  koin, anda bisa menukar  $N$  koin  $Z$  menjadi  $N/2$ ,  $N/3$ , dan  $N/4$  koin  $Z$ . Anda dapat menukar koin  $Z$  menjadi koin  $X$  dengan perbandingan  $1 : 1$ .

Berapa banyak koin  $X$  maksimal yang bisa didapatkan?

Constraint :  $2 \leq N \leq 10^9$

# DP PAKAI MAP

State dan transisi nya :

State nya N yaitu koin Z yang kita punya sekarang.

Jadi base case nya adalah  $DP(0)=0$

transisi nya adalah :  $DP(N) = \max(N, DP(N/2)+DP(N/3)+DP(N/4))$

# DP PAKAI MAP

memori yang digunakan hanya sebanyak  $\log_2(N) + \log_3(N) + \log_4(N)$   
untuk setiap pemanggilan  $DP(N)$

# State Reduction

Ada  $N$  barang  $X$  dan  $M$  barang  $Y$ , nilai barang  $X$  adalah  $K$  sedangkan nilai barang  $Y$  adalah  $J$ .

Sekarang anda memiliki karung spesial, karung ini dapat mengubah nilai barang yang dimasukkan dengan formula  $v = \text{nilai\_barang} * S$  dimana  $S$  adalah sisa ukuran karung.

Diberikan Karung dengan ukuran  $C$  dan ukuran setiap barang  $N$  dan  $M$  carilah nilai maksimal yang dapat dihasilkan

# State Reduction

Constraint

$$1 \leq N, M \leq 2000$$

$$1 \leq K, J, C \leq 10^7$$

$$1 \leq X_i, Y_i \leq 10^7$$

# State Reduction

Bisa pakai Knapsack

State :

$N, M, C$

total state  $N * M * C = 2000 * 2000 * 10^7 = \text{MLE}$

# State Reduction

Pakai Map? bisa MLE atau TLE duluan

Flying Table? masih MLE



# State Reduction

Sorting dulu barang-barangnya. Kenapa?

Karena akan paling untung jika kita mengurangi ukuran karung sesedikit mungkin agar perkaliannya maksimal.

# State Reduction

Tapi masih perlu memperhatikan kedua macam barang.

Tadi masih MLE, tapi sekarang bisa di DP. Kenapa?

# State Reduction

Karena kita tidak perlu mengingat sisa ukuran karung, dikarenakan bagaimanapun cara pengambilan barang untuk state  $(x,y)$  akan selalu menghasilkan sisa ukuran karung yang sama, sehingga kita hanya perlu memori  $N*M=2000*2000 = 4000000$

# GREEDY

Tergantung Observasi.

Biasanya (tidak semuanya) :

- Sorting pakai comparing spesial (atau sorting biasa)
- Coba jalan dari belakang ke depan
- ditambah DS

# GREEDY

Ada  $N$  jenis barang dengan kuantitas infinite, kita punya budget  $M$ .

Tiap jenis barang punya harga  $H_i$  dan tingkat kepuasan  $F_i$ .

Tentukan total tingkat kepuasan terbesar yang bisa dicapai

$$1 \leq N \leq 10^5$$

$$1 \leq M \leq 10^5$$

$$1 \leq H_i, F_i \leq 10^5$$

$H_i$  habis membagi  $H_j$  atau sebaliknya untuk setiap  $(i, j)$

# GREEDY - SORTING

Sort berdasarkan  $F_i / H_i$  menurun

Beli gila-gilaan secara greedy (beli selama budget masih ada)

# GREEDY

Kita punya kayu sepanjang  $M$ , dan mau bikin  $N$  potong kayu

Potongan ke- $i$  panjangnya  $p_i$ .

Biaya motong dari kayu dengan panjang  $X$  itu sama dengan  $X$

Tentukan biaya minimum untuk bikin  $N$  potong kayu!

Jumlah semua  $p_i \leq M$

$1 \leq N, M \leq 10^5$

$1 \leq p_i \leq M$

$\text{sum}(p_i) = M$



# GREEDY - PQ

PQ dari hasil pemotongan akhir

Selalu gabungin dua potongan yang ukurannya paling pendek

# GREEDY

Ada  $N$  buah semut, semut ini akan memakan semut yang ada disekitarnya jika semut sekitarnya lebih kecil ukurannya dari dirinya.

Diberikan ukuran setiap semut,

Tentukan Maksimum semut yang dimakan tiap semut.

$$1 \leq N \leq 10^5$$

# GREEDY - STACK

Kompleksitas  $O(N)$

kita melakukan iterasi dari semut paling kiri hingga paling kanan.

untuk setiap langkah, selama stack tidak kosong dan data teratas stack lebih kecil sama dengan data sekarang

catat jangkauan terjauh semut yang ada di stack =  
(index\_sekarang-index\_semut\_teratas)

# GREEDY-STACK

dari langkah tadi kita mendapatkan jangkauan maksimum kearah kanan, bagaimana cara untuk mendapatkan jangkauan maksimum kearah kiri?

# GREEDY - STACK

jika stack masih ada setelah iterasi selesai,  
untuk setiap semut yang ada di stack, dicatat jangkauannya dengan  
menanggapi index sekarang adalah  $N+1$

# GREEDY - STACK

Lakukan iterasi dari kanan ke kiri, caranya sama

A decorative graphic on the left side of the slide, consisting of a network of light blue lines and small circles, resembling a circuit board or a stylized tree structure, set against a blue gradient background.

THANK YOU!