

# Hand Out Dynamic Programming

## 1. Quick Sum (QS)

$\rightarrow 1D$ : ရှုံး array အား  $A[1 \sim N]$  (ကြည့်မှတ်နိုင်) ပေါ်  $[a, b]$  ပုံမှန်လိုက်ရမှု?  $\leftarrow$  မျှော် Q မျှော်.

sol: ပါ ရှုံးပုံမှန်  $\rightarrow O(NQ)$

sol: ပို့ကဲ  $\rightarrow$  တိမ်စွာပုံမှန်လိုက်မှတ်ဆောင်ခြင်း.

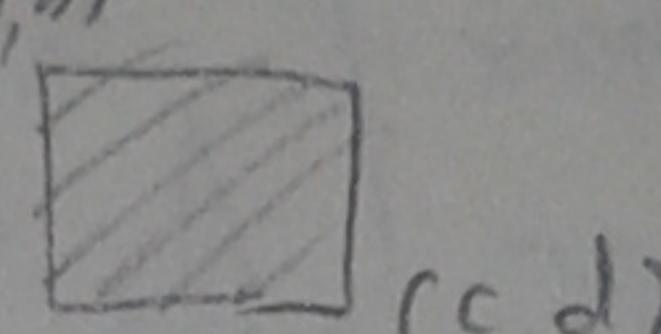
$$dp[i] = \sum_{a=1}^i A_a \rightarrow O(N) \quad \begin{bmatrix} \text{base case} \\ dp[0] = 0 \end{bmatrix}$$

ပုံမှန်လိုက်  $\sum A_a \sim A_b$  ရှိနိုင်သူ  $dp[b] - dp[a-1]$   $\rightarrow O(1)$

စုစုပေါ်  $O(N) + O(Q)$

$\rightarrow 2D$ , ရှုံးမှတ်သွေး  $N \times M$  (ကြည့်မှတ်နိုင်)

ပုံမှန်  $\square$  မှတ်ဆောင်



ပုံမှန်လိုက်ရမှု.  $\leftarrow$  မျှော် Q မျှော်

idea ရှုံး  $dp[i][j] = \sum$   $\rightarrow O(NM)$

ပုံမှန်လိုက်ရမှုလုပ်လမ်း

$$\boxed{(a,b)} = \boxed{\phantom{dp}} - \boxed{\phantom{dp}} - \boxed{\phantom{dp}} + \boxed{\phantom{dp}} \quad \text{မျှော် } O(1)$$

$$\boxed{(a,b)} = \boxed{dp(c,d)} - \boxed{dp(a-1,d)} - \boxed{dp(c,b-1)} + \boxed{dp(a-1,b-1)}$$

စုစုပေါ်  $O(NM) + O(Q)$

? - ပုံမှန်လိုက်  $k$  ကို  $\rightarrow$  big-Oh ပုံမှန်လိုက်

- ပုံမှန်လိုက် အတွက်  $\sim 100M$  ပုံမှန်လိုက် ပုံမှန်လိုက် အတွက်  $\sim 100M$

! update ပုံမှန်လိုက်  $\rightarrow$  ပုံမှန်လိုက် (binary index tree)

## Q Max Sum (MS)

Q: ចាំនួន  $A[1..n]$

តួនាទី (លើកដឹងទុក)

Sol នេះ  $\rightarrow$  loop  $\sum$  ស្ថិតិមែន កំណត់តាមលក្ខណៈ  
ក្នុងបញ្ហាអារីម៉ូណុយ

$\hookrightarrow$  ស្ថិតិមែន QS  $\rightarrow O(N^2)$

$\hookrightarrow$  ស្ថិតិមែន  $\rightarrow O(N^3)$

Sol dp  $\rightarrow$  ជីថល state

$dp(i)$ : MS  $A[1..i]$  អីដែល  $A_i$  ធ្លាត់ពេលវេលាដែល MS

transition

$$dp(i) = \max \begin{cases} A_i \\ A_i + dp(i-1) \end{cases}$$

$$dp(0) = 0.$$

ការណែនាំ ស្ថិតិមែន Array dp.  $\rightarrow O(N)$

print sol

ស្ថិតិមែន  $dp(x)$  ត្រូវបាន

(មិនត្រូវ ទិន្នន័យ  $A_x$  តាមព័ត៌មាននៃលទ្ធផល)  $\rightarrow$  បានស្ថិតិមែន  $A_x$  និង  $dp(x)$  ត្រូវបាន

បានស្ថិតិមែន  $A_x$  និង  $dp(x)$  ត្រូវបាន

Q+: ចាំនួន នូវ  $N \times M$

? បានដោយ  $\square$  មុនការ និង  $dp$  រួចរាល់ 1

8

\* តាម MS, QS

ស្ថិតិមែន bottom up

108:

EXTR

### 3. Longest Increasing Subsequence (LIS)

subsequence → ចំណាំកែតាមការពិនិត្យ

substring → ចំណាំអាជីវកម្ម

Q: ត្រូវការណា  $A[1..n]$  ជា subsequence ដើម្បីបានអាចត្រួតពិនិត្យ<sup>1</sup>  
ទិន្នន័យលើរបាយការណ៍ ឬសម្រាប់ការពិនិត្យ / តាមរបាយការណ៍

Sol dp:  $dp(i) = \text{LIS } A[1..i]$  ដើម្បី  $A_i$  ដែលមានការពិនិត្យ LIS

$$dp(i) = \max \left\{ dp(j) \mid j < i \text{ និង } A_j < A_i \right\} + 1$$

នៅលើ: នៅលើ  $A_i$ , យុទ្ធសាស្ត្រ

→ ការរាយការណ៍ រាយការណ៍  $dp(j)$  ឡើង

+ 1 ជាសម្រាប់ការពិនិត្យ / ការរាយការណ៍  $dp(j)$  → set និង 1

តាមរបាយការណ៍  $dp$   
→  $O(N^2)$ , inner state loop  $O(N)$ , &  $O(N)$  state

print sol: ត្រូវការពិនិត្យសរុប → print recursive

∴ Sol អាជីវកម្ម →  $O(N \lg N)$ . (upper-bound)

problem:

10684: MS 1D ត្រូវបានការពិនិត្យ  $\Sigma > 0$  ដោយ Losing streak.

507: MS 1D នូវ print រាយការណ៍ sol ដែលបានពិនិត្យដោយប្រើប្រាស់ recursion

នូវការពិនិត្យ / គូនុញ្ញនឹងការពិនិត្យដោយប្រើប្រាស់

• ឧប្បរស  $\star$  test case, ឬនៃពេល  $N$  មានតូចរាល់  $N-1$  នៅ

1	2	3
-1	6	

108: MS 2D  $N^4$  រឿង / 10827: MS 2D ឱ្យបានស្ថាបន្ទាត់!

EXTRA: foot / c2 ox racing

## 4. Longest Common Subsequence (LCS)

Q: ស្នើសារ  $A[1..n]$ ,  $B[1..m]$

និង subsequence  $S[1..k]$  ដែលជា subsequence នៃ  $A$  និង  $B$

Sol dp:  $dp(i, j) = \underset{\text{def}}{\text{LCS}} A[1..i], B[1..j]$

$$dp(i, j) = \begin{cases} \text{LCS}(i-1, j-1) + 1 & ; A[i] = B[j] \\ \max\{\text{LCS}(i-1, j), \text{LCS}(i, j-1)\} & ; A[i] \neq B[j] \end{cases}$$

base case: 0 if  $i=0$  or  $j=0$

ឧបាទ់ → ចំណាំរាយការណ៍ → +1 ឱ្យលើកវាតិតាត់រាយការណ៍

→ មុនឯកសារ → ត្រូវការពាក្យសារការណ៍ និង  $\Theta(NM)$

→ ភាពិត្យសារការណ៍ និងនិរន័យនឹង subsequence.

print sol ① ត្រូវបានបញ្ជាកេណៈ បញ្ជាកេណៈ  $dp(N, M)$  នៅ.

② ដោយត្រូវ  $dp(i, j) * A[i] = B[j] ?$ .

ការ (i) → ឱ្យលើកវាតិតាត់  $A[i]$  →  $i--$ ,  $j--$

ការ (ii) → ឱ្យលើកវាតិតាត់  $B[j]$  →  $i--$ ,  $j--$  (ត្រូវការពាក្យ)

លទ្ធផល: long fox

Ex TRA 10B1 - ស្នើសារ  $\Sigma$  col → ត្រូវបានបញ្ជាកេណៈ និង ត្រូវបានបញ្ជាកេណៈ និង  $dp(N, M)$  / ត្រូវបានបញ្ជាកេណៈ និង  $dp(N, M)$  / ត្រូវបានបញ្ជាកេណៈ និង  $dp(N, M)$

10192-LCS / lcs ឬ grader (ឬ substring)

\* POJ-1159 និងការសម្រេចការណ៍ និង string និង word នៃលោកស្រី

Ex 335B - ឬ palindrom នាមឈាន 100 នាមឈាន string

(~~palindrom~~ និងការណ៍)

និងការណ៍ និង

5. Q1

Q1

sol dp

+ និងការណ៍

dp

ប្រា

របាយ

print

~~លទ្ធផល~~

រ. ផ្លូវ

V. ទាន់

R. និង

G. ឱ្យលើ

សម្រាប់



## 5. 01 - knapsack

Q សេវានិភ័យ និង ការពារណា និង របៀប រៀបចំ និង រៀបចំ សេវានិភ័យ

ရှေ့ချောင်းမြတ်သွေးလုပ်လုပ်မှုများ  
(ရှေ့ချောင်းမြတ်သွေးလုပ်လုပ်မှုများ)

Sai  $d_p$ :  $d_p(i, j) = \text{distance}_{\text{Manhattan}}(i, j)$   
between  $v_0, v_i$  and  $v_0, v_j$

“សេរីអារ៉ា តាមឈានឃុំ អាសយដ្ឋានកំណើនខេត្តពោធិ៍សាស្ត្រ  
និងការងារ និងការបង្កើត និងការប្រើប្រាស់ និងការប្រើប្រាស់”

$$dp(i, j) = \max \left\{ \begin{array}{l} dp(i-1, j) \\ dp(i-1, j - w_i) + v_i \end{array} \right.$$

→ A  
← B

→ O(NW)

$$\lim_{n \rightarrow \infty} \frac{a_n}{b_n} < 0 \rightarrow -\infty \quad ; \quad \lim_{n \rightarrow \infty} b_n \rightarrow 0 \rightarrow 0.$$

Although  $D_p(N, n)$  two

ବେଳେ ଅପରାଧ କରିବାକୁ ଆମେ କାହାର କାହାର କାହାର କାହାର କାହାର

print sal → ගැන මුදලන් යුතු  $\textcircled{A}$  වේ  $\textcircled{B}$  මුදලන් යුතු

~~old~~ front knapsack

ရန်ကုန်မြို့၊ မန္တလေးမြို့၊ မန္တလေးမြို့၊ မန္တလေးမြို့၊ မန္တလေးမြို့

Q. నేను 9వ శాఖలోనుగోళకు 9.05 త్రయిల్ రూపులు కనుగొని to) - toy story

Some trick 1D solve knapsack

