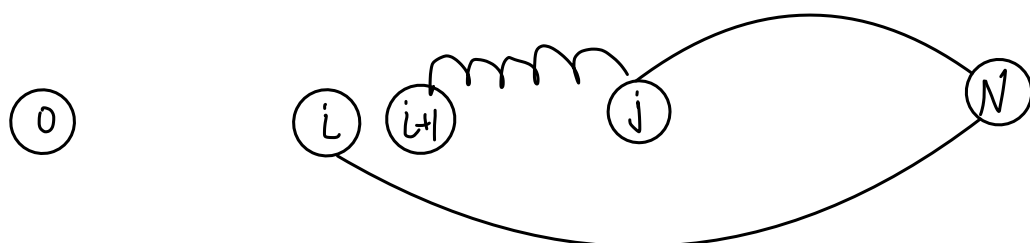


5. dp is symmetric, since $E(a,b) = E(b,a)$

(a)

$$dp(i, j) = \begin{cases} \text{Null} & , i = j \\ dp(j, i) & , i > j \\ E(0, j) & , i = 0, j = 1 \\ dp(i, j-1) + E(j-1, j) & , i+1 < j \\ \min_{0 \leq k < i} (E(k, j) + dp(k, k+1) + \sum_{k < l < i} E(l, l+1)) & , \text{else } (i+1 = j) \end{cases}$$

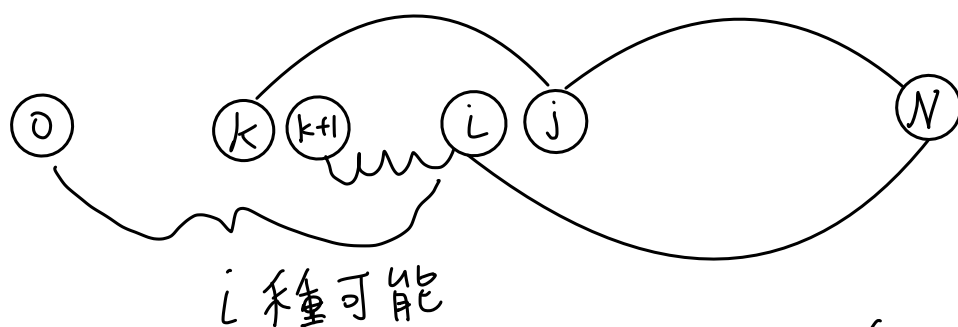
Case 1: $j - i > 1$



由於 $i+1$ 到 j 每一步都要走

所以從 $dp(i, i+1)$ 開始往後每步算一次就好

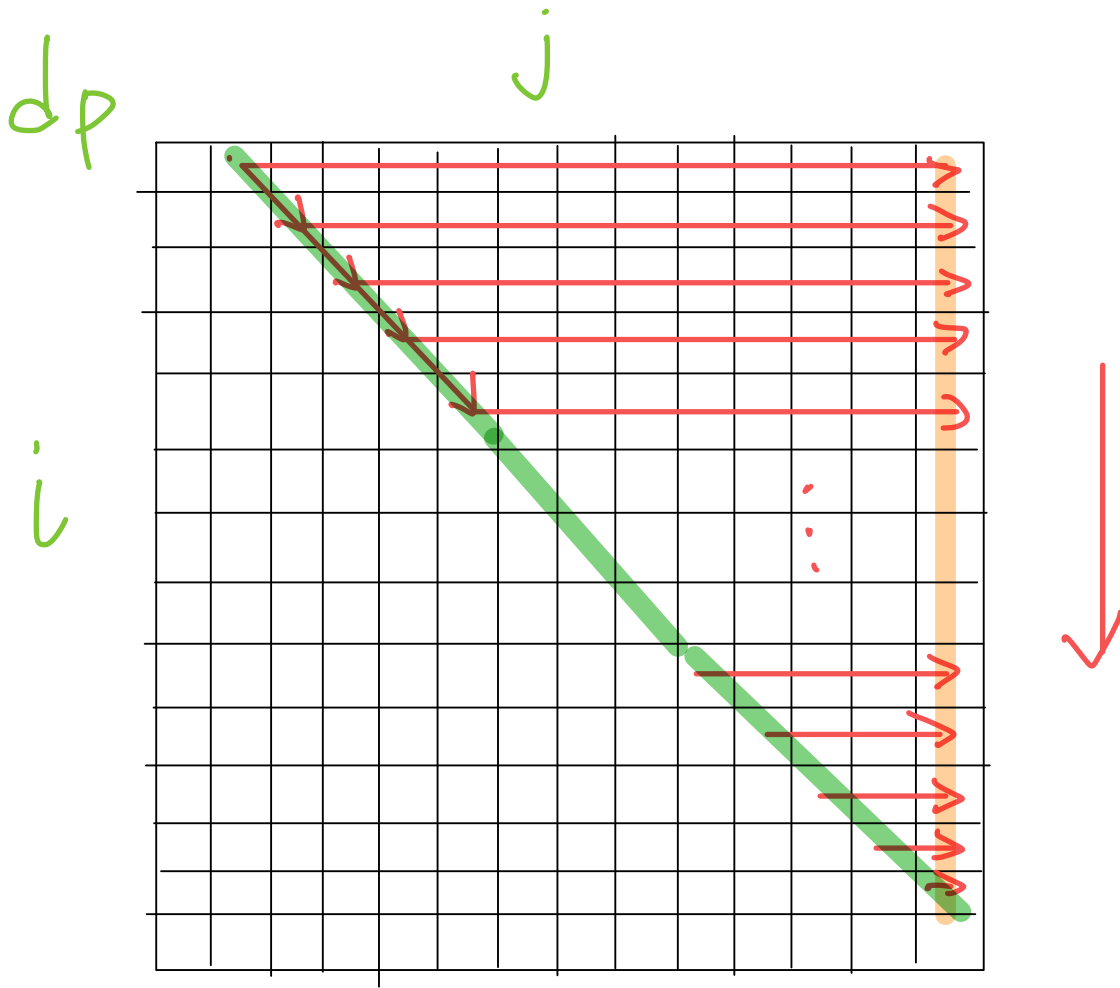
Case 2: $j - i = 1$



找出 j 的前一步 k , 使得 $(E(j, k), i \text{ 到 } k+1 \text{ 每步都走, 加上 } dp(k, k+1))$ 最小的 k 就好

($i \sim k+1$ 的 sum 可以使用 prefix sum 的方法
讓每次都是 $O(1)$)

Time & Space complexity

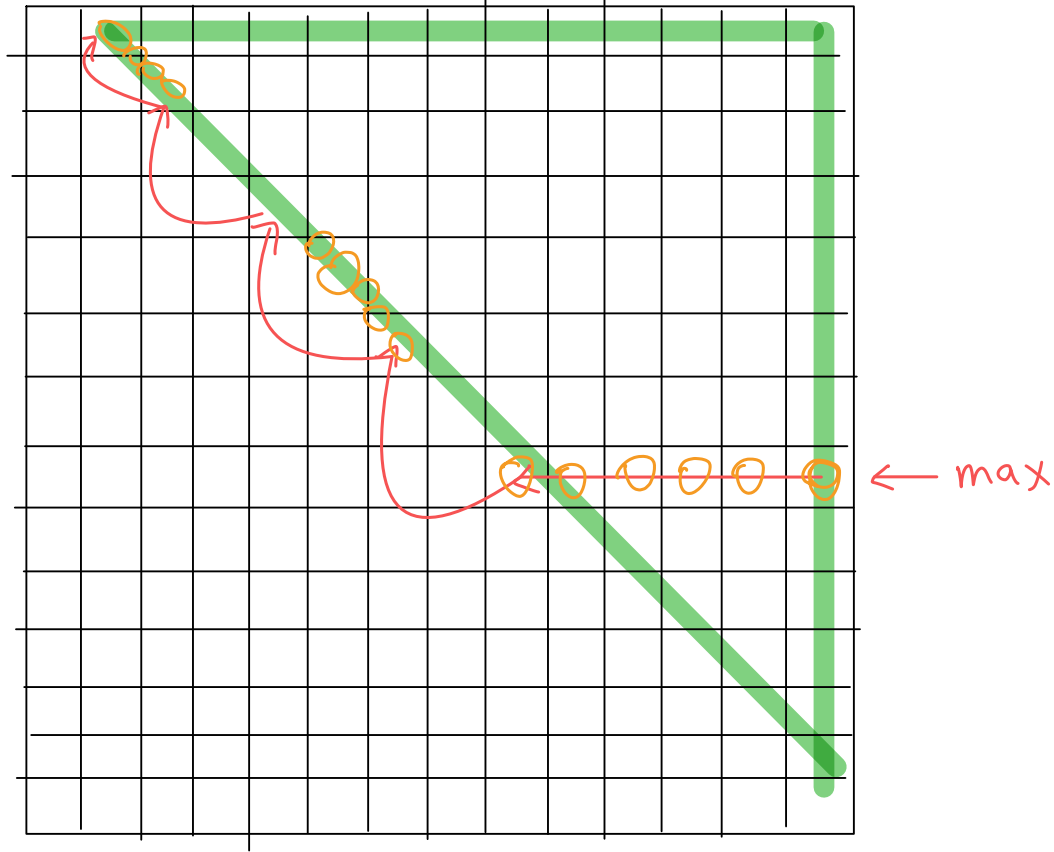


每一層需 i 的時間找 k
， $N-i$ 的時間掃過整層

$$(i+N-i) \times N \text{ 層} = O(N^2) \text{ for time complexity}$$

(2) 只需要存 ● ● 所畫的部份, 其它者不會在後面用到 $\Rightarrow O(N)$ for space complexity

(3) 紀錄每次 case 2 找到的 k
再由以下路徑, 每單數的箭頭把經過的點加入 S_{go} (前往 N 的路) 其它的為 S_{left}



(b)

(1) Let $\text{Sum}_h = \sum D_n$

(2) $dp(i, j, h) =$

$$\left\{ \begin{array}{ll} \text{Null} & , i = j \\ dp(j, i, \text{Sum}_h - h) & , i > j \\ E(0, j) & , i = 0, j = 1 \\ dp(i, j-1, h - D_j) + E(j-1, j) \text{ if } D_j < h, \text{ else } -1 & , i+1 < j \\ \min_{0 \leq k < i, D_k < h} (E(k, j) + dp(k, k+1, h - D_k) + \sum_{k < l < i} E(l, l+1)) & , i+1 = j \\ -1 & , h \leq 0 \end{array} \right.$$

只需在原DP上每個點紀錄H個值就好, Time complexity $O(HN^2)$

Space complexity $O(HN)$

6.
(1) 98141210 under no assumption
981120 under assumption 3

(2) 由大到小串接 (排序 $O(n \log n)$)

(3) 由大到小串接 (比較高 \rightarrow 低位,
若比完其中一位數較少的, 將它重覆一次再
繼續比。 eg. $\begin{matrix} 123 \\ 123123124 \end{matrix} \checkmark$ $\begin{matrix} 123 \\ 123123122 \end{matrix} \checkmark$

(4) 統計 $\text{mod } 3 = 0, 1, 2$ 個數 in nums
分成 3 個 arr $\frac{n}{3}$

各別用 bucket sort 排好

從 arr1, arr2 丟掉最小的 $\text{abs}(\text{arr}[i] - \text{arr}[j]) \% 3$ 個

剩下再用 bucket sort 由大到小串接起來。

(5) 98653

(6)

$$\text{define } a \oplus b = \begin{cases} a \times 10^{\text{digit of } b} + b, & \text{else} \\ \text{null} & , a = \text{null or } b = \text{null} \end{cases}$$

$$P_1 \sim P_n \quad M_1 \sim M_n$$

maximum satisfying value for

$$dp(i, j, k) = P_i \sim P_n, M_j \sim M_n, \text{ at most } k \text{ courses}$$

$$\begin{cases} \max(P_i, M_j, dp(i+1, j, k), dp(i, j+1, k)) & , k=1 \\ \max(P_i \oplus dp(i+1, j, k-1), M_j \oplus dp(i, j+1, k-1), dp(i+1, j, k), dp(i, j+1, k)) & , k > 1 \text{ and } i \leq n \text{ and } j \leq n \\ \text{null} & , (i > n \text{ or } j > n) \text{ and } k > 1 \end{cases}$$

e.g. $P_i = [3, 4, 6, 5, 0]$, $M_i = [9, 0, 5, 8, 3]$, $k=5$

$k=1$

	3	4	6	5	0
9	9	9	9	9	9
0	8	8	8	8	8
5	8	8	8	8	8
8	8	8	8	8	8
3	6	6	6	5	3

and it's clear

that (we fill $n \times n$ table and $O(1)$ for each slot) for k times

$$\Rightarrow O(kn^2)$$

$k=2$

	3	4	6	5	0
9	98	98	98	98	98
0	86	86	86	85	83
5	86	86	86	85	83
8	86	86	86	85	83
3	65	65	65	53	30

$k=3$

	3	4	6	5	0
9	986	986	986	985	983
0	853	853	853	853	830
5	853	853	853	853	830
8	865	865	865	853	830
3	653	653	653	530	null

$$k=4$$

	3	4	6	5	0
9	9865	9865	9865	9853	9830
0	8653	8653	8653	8530	5830
5	8653	8653	8653	8530	5830
8	8653	8653	8653	8530	null
3	6530	6530	6530	null	null

$$k=5$$

	3	4	6	5	0
9	98653	98653	98653	98530	95830
0	86530	86530	86530	58530	05830
5	86530	86530	86530	58530	null
8	86530	86530	86530	null	null
3	46530	46530	null	null	null