



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Thực hành TTUĐ

Week 4

Dynamic Programming

1. Even Subsequence

2. Gold Mining

3. Warehouse

4. Nurse

1. Max even subsequence

Đề bài:

Cho một dãy số a gồm n số nguyên a_1, a_2, \dots, a_n .

Một đoạn con được định nghĩa là một đoạn các phần tử liên tiếp của dãy a .

Yêu cầu:

Tìm đoạn con có trọng số là chẵn và có tổng các phần tử là lớn nhất.

Input	Output
8 4 -5 2 4 -8 2 3 1	6

Đầu vào được diễn giải như sau

- Dòng 1 là số lượng phần tử trong dãy ban đầu n , $n < 10^6$
- Dòng 2 lần lượt là giá trị các phần tử trong dãy, với giá trị trong khoảng $[-10^6, 10^6]$

Tìm và in ra màn hình đoạn con có tổng chẵn lớn nhất (nếu có) trong dãy n phần tử trên. Trong trường hợp KHÔNG tồn tại đoạn con tổng chẵn thì in ra là NOT_FOUND (VD. Dãy chỉ gồm 1 số lẻ)

Gợi ý:

- Đoạn con trong khoảng i tới j sẽ là các phần tử nằm liên tiếp nhau: $a_i, a_{i+1}, a_{i+2}, \dots, a_j$
- Tổng giá trị của đoạn con là tổng giá trị các phần tử $a_i + a_{i+1} + a_{i+2} + \dots + a_j$
- Nếu vét cạn số lượng đoạn con chẵn trong dãy n phần tử thì số lượng phải xét sẽ là $O(n^2)$, nếu n tới cỡ 1tr thì vẫn có thể vét được, nhưng mà thời gian sẽ khá lâu, tính tới phút

Max even subsequence – Dynamic Programming Algorithm

- Xác định bài toán con
 - $S0[i]$ là tổng của dãy con **chẵn** cực đại của dãy $a[1], \dots, a[i]$ mà phần tử cuối cùng là $a[i]$
 - $S1[i]$ là tổng của dãy con **lẻ** cực đại của dãy $a[1], a[2], \dots, a[i]$ mà phần tử cuối cùng là $a[i]$
- Điều kiện đầu
 - Nếu $a[1]$ chẵn thì $S0[1] = a[1]$ và $S1[1]$ không tồn tại
 - Nếu $a[1]$ lẻ thì $S1[1] = a[1]$ và $S0[1]$ không tồn tại
- Công thức quy hoạch động
 - Với mỗi i từ 2 đến n : tính đồng thời $S0[i]$ và $S1[i]$ phụ thuộc $S0[i-1]$, $S1[i-1]$, $a[i]$ tùy thuộc vào $a[i]$ chẵn hay lẻ.

2. Gold Mining

- The Kingdom ALPHA has n warehouses of golds located on a straight line and are numbered $1, 2, \dots, n$. The warehouse i has amount of a_i (a_i is non-negative integer) and is located at coordinate i ($i = 1, \dots, n$). The King of ALPHA opens a competition for hunters who are responsible to find a subset of gold warehouses having largest total amount of golds with respect to the condition that the distance between two selected warehouses must be greater than or equal to $L1$ and less than or equal to $L2$.
- **Input**
 - Line 1 contains n , $L1$, and $L2$ ($1 \leq n \leq 100000, 1 \leq L1 \leq L2 \leq n$)
 - Line 2 contains n integers a_1, a_2, \dots, a_n
- **Output**
 - Contains only one single integer denoting the total amount of golds of selected warehouses.
- **Example:**
- **Input**
6 2 3
3 5 9 6 7 4
- **Output**
19

2. Gold Mining – Backtracking Algorithm

- Duyệt hết tất cả các trường hợp chọn các nhà kho khác nhau:
- Với mỗi trường hợp, kiểm tra xem 2 nhà kho liên tiếp có khoảng cách nằm trong khoảng $[L1, L2]$ hay không, nếu tất cả các nhà kho đều thỏa mãn thì cập nhật tổng lượng vàng.
- Độ phức tạp: $O(2^n * n)$.
- Có thể thực hiện một số biện pháp nhánh cận như:
- Khi đang xét đến nhà kho thứ i , cần nhắc chỉ xét các nhà kho trong đoạn $[i + L1, i + L2]$.

2. Gold Mining

Đề bài:

Có n nhà kho nằm trên một mặt phẳng.

Nhà kho i có số lượng vàng là a_i .

Yêu cầu:

Chọn các nhà kho sao cho:

Tổng lượng vàng là lớn nhất.

2 nhà kho liên tiếp có khoảng cách nằm trong khoảng $[L1, L2]$.

2. Gold Mining – Dynamic Programming Algorithm $O(N^2)$

- Gọi $F[i]$ là tổng lượng vàng lớn nhất nếu chọn các nhà kho từ 1 đến $i-1$ và nhà kho thứ i được chọn.
- Khởi tạo: $F[i] = a[i]$.
- Công thức:

$$F[i] = \max_{j \in [i-L2, i-L1]} (a[i] + F[j]), \forall i \in [L1, n].$$

- Kết quả:

$$\max_i F[i], \forall i \in [1, n].$$

Độ phức tạp: $O(N^2)$.

2. Gold Mining – Dynamic Programming Algorithm ($O(n)$)

- Hàng đợi 2 đầu (deque) là cấu trúc dữ liệu kết hợp giữa hàng đợi và ngăn xếp -> phần tử đều có thể được thêm vào và lấy ra ở đầu và ở cuối deque.
- Thao tác: `push_back()`, `push_front()`, `pop_back()`, `pop_front()`
- Cải tiến: Các phần tử trong hàng đợi là chỉ số j tham gia vào ứng viên xác định $F[i]$.
 - Duyệt $F[i]$ theo thứ tự $i = 2, 3, \dots, n$.
 - Xóa mọi phần tử j mà $F[j] \leq F[i - L1]$ trong hàng đợi, thêm chỉ số $i - L1$ vào hàng đợi.
 - Xóa phần tử đầu tiên `top` của hàng đợi cho đến khi $\text{top} \geq i - L2$.
 - $F[i] = F[\text{top}] + a[i]$.

3. Warehouse

- A truck is planned to arrive at some stations among N stations $1, 2, \dots, N$ located on a line. Station i ($i = 1, \dots, N$) has coordinate i and has following information
 - a_i : amount of goods
 - t_i : pickup time duration for taking goods
- The route of the truck is a sequence of stations $x_1 < x_2 < \dots < x_k$ ($1 \leq x_j \leq N$, $j = 1, \dots, k$). Due to technical constraints, the distance between two consecutive stations that the truck arrives x_i and x_{i+1} is less than or equal to D and the total pickup time duration cannot exceed T . Find a route for the truck such that total amount of goods picked up is maximal.
- **Input**
 - Line 1: N, T, D ($1 \leq N \leq 1000$, $1 \leq T \leq 100$, $1 \leq D \leq 10$)
 - Line 2: a_1, \dots, a_N ($1 \leq a_i \leq 10$)
 - Line 3: t_1, \dots, t_N ($1 \leq t_i \leq 10$)
- **Output**
 - Write the total amount of goods that the truck picks up in the route.



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

3. Warehouse – Dynamic Programming Algorithm

- Xác định bài toán con
 - Gọi $f[i][k]$ là số lượng hàng lớn nhất có thể lấy được nếu chỉ xét những nhà kho từ 1 $\rightarrow i - 1$, lấy nhà kho thứ i và thời gian lấy hàng không vượt quá k .
- Công thức quy hoạch động:
 - If $k < t[i]$: $f[i][k] = 0$;
 - If $k \geq t[i]$: $f[i][k] = \max(f[j][k - t[i]] + a[i]), j = i - D, \dots, i - 1$;
- Kết quả:
 - $\max(f[i][k]), i = 1, \dots, n, k = 1, \dots, T$;
- Độ phức tạp:
 - $O(n * T * D)$.

4. Nurse

- The director of a hospital want to schedule a working plan for a nurse in a given period of N consecutive days $1, \dots, N$. Due to the policy of the hospital, each nurse cannot work all the days $1, \dots, N$. Instead, there must be days off in which the nurse need to take a rest. A working plan is a sequence of disjoint working periods. A working period of a nurse is defined to be a sequence of consecutive days on which the nurse must work and the length of the working period is the number of consecutive days of that working period. The hospital imposes two constraints:
 - Each nurse can take a rest only one day between two consecutive working periods. it means that if the nurse takes a rest today, then she has to work tomorrow (1)
 - The length of each working period must be greater or equal to $K1$ and less than or equal to $K2$ (2)
 - The director of the hospital want to know how many possible working plans satisfying above constraint?
- **Input**
 - The input consists of one line which contains 3 positive integers $N, K1, K2$ ($2 \leq N \leq 1000, K1 < K2 \leq 400$)
- **Output**
 - The output consists of only one single integer M modulo 10^9+7 where M is the total working plans satisfying the above constraints.

4. Nurse

Đề bài:

Một giám đốc của một bệnh viện muốn lập lịch làm việc cho y tá trong N ngày $1 \dots N$.

Y tá chỉ có thể làm việc liên tục trong x ngày ($K1 \leq x \leq K2$), sau đó phải nghỉ 1 ngày.

- Y tá chỉ được nghỉ 1 ngày giữa 2 đợt làm việc liên tiếp. Có nghĩa rằng nếu y tá nghỉ ngày hôm nay, ngày mai y tá đó sẽ phải làm việc.

Yêu cầu:

Tính số cách có thể lập lịch làm việc cho các y tá.

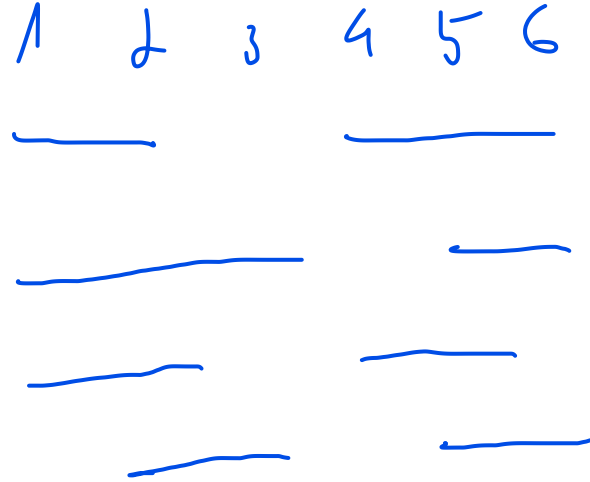
Example:

Input:

6 2 3

Output:

4



Nurse – Dynamic Programming Algorithm

- Xác định bài toán con:
 - Gọi $S0[i]$ là số cách lập lịch đến ngày thứ i và ngày thứ i là ngày nghỉ.
 - Gọi $S1[i]$ là số cách lập lịch đến ngày thứ i và ngày thứ i là ngày làm việc.

→ số cách lập

- Khởi tạo:
 - $S0[i] = S1[i] = 0$, với mọi $i = 1, \dots, n$
 - $S0[1] = 1, S1[1] = 1, S0[0] = 1$

→ bước \Rightarrow số cách là $\boxed{11}$ đôi lịch
thời đi đầu

- Công thức quy hoạch động:

$$S0[i] = S1[i-1];$$

$$S1[i] = \sum_{j=k_1}^{k_2} S0[i-j];$$

- Kết quả:

$$S0[n] + S1[n];$$

- Độ phức tạp:

$$O(n^2).$$

		k_1	k_2				
		\uparrow	\uparrow				
	0	1	2	3	4	5	6
$S0$	1	1	0	0	0	0	0
$S1$	0	1	0	0	0	0	0

→ for ($i = 2 \rightarrow n$) $\rightarrow i = 2 \Rightarrow$

$$\begin{cases} S0[i] = S1[i-1] \\ S1[i] = \sum_{j=k_1}^{k_2} S0[i-j] \end{cases}$$

$$= \sum_{j=k_1}^{k_2} S0[i-j]$$

$$= S0[2-2] + S0[2-3]$$