

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.

Đề bài:

N nhà kho được đặt tại các vị trí từ 1 đến N. Mỗi nhà kho có:

a_i là số lượng hàng.

t_i là thời gian lấy hàng.

Tìm một tuyến đường lấy hàng đi qua các trạm $x_1 < x_2 < x_3 < \dots < x_k$ ($1 < x_j < N, j = 1 \dots k$) sao cho:

$$x_{i-1} - x_i \leq D \quad \forall i \in [1, k].$$

$$t_1 + t_2 + \dots + t_k \leq T.$$

$$a_1 + a_2 + \dots + a_k \text{ lớn nhất.}$$

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.

Warehouse

Example:

Input:

6 6 2

6 8 5 10 11 6

1 2 2 3 3 2

Output:

24

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 -> $i - 1$, 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:
 - $O(n * T * D)$.

Implementation – initialization

```
#include <bits/stdc++.h>

using namespace std;

const int N = 1e3 + 2;
int n, f[N][102], T, D, a[N], t[N], res;

void inp()
{
    ios_base :: sync_with_stdio(0); cin.tie(NULL);
    cin >> n >> T >> D;
    for(int i = 1 ; i <= n; i ++) cin >> a[i];
    for(int i = 1 ; i <= n; i ++) cin >> t[i];
}
```

Implementation – main process

```
void proc()
{
    for(int i = 1 ; i <= n; i ++) {
        for(int k = t[i]; k <= T; k++) {
            for(int j = max(0, i - D); j <= i - 1; j ++) {
                f[i][k] = max(f[i][k], f[j][k - t[i]] + a[i]);
            }
            res = max(res, f[i][k]);
        }
    }
    cout << res << "\n";
}

int main()
{
    inp();
    proc();
}
```