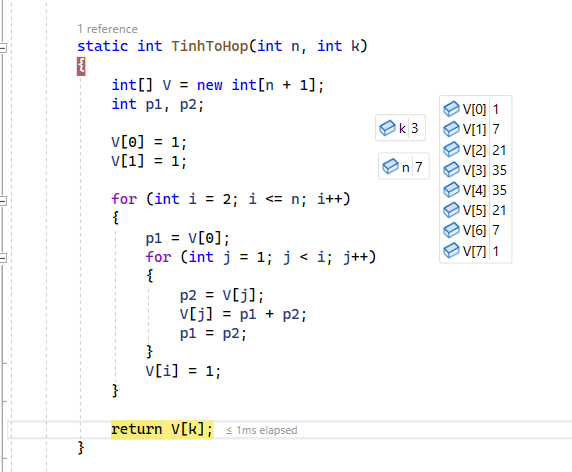
**Tổ hơp**

Kết quả:

A screenshot of a computer

Description automatically generated



**Khởi tạo mảng V: O(n)**

Tạo mảng kích thước n + 1.

**Khởi tạo giá trị cho V[0] và V[1]: O(1)**

Gán giá trị cố định.

**Vòng lặp ngoài (i từ 2 đến n): O(n)**

Lặp n - 1 lần.

**Vòng lặp trong (j từ 1 đến i - 1): O(n^2)**

Tổng số phép lặp là tổng của 1 + 2 + ... + (n - 1), dẫn đến O(n^2).

**Cập nhật giá trị V[j] và V[i]: O(1)**

*Các phép toán cộng và gán.*

**Trả về V[k]: O(1)**

Truy cập và trả về phần tử.

**Balo**

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Khởi tạo mảng F và X: O(soLuong \* W)**

Tạo hai mảng kích thước (soLuong + 1) x (W + 1).

**Tính toán bảng F và X (Hàm TaoBang):**

Vòng lặp ngoài (i từ 2 đến soLuong): O(soLuong)

Vòng lặp trong (v từ 0 đến W): O(W)

Vòng lặp trong (xk từ 1 đến yk): O(W/TrongLuong[i]), tính tổng: O(W)

Tổng: O(soLuong \* W)

**Truy vết bảng (Hàm TraBang):**

Vòng lặp (i từ soLuong đến 1): O(soLuong)

Tổng: O(soLuong)

**In kết quả và tính tổng giá trị:**

Vòng lặp (i từ 1 đến soLuong): O(soLuong)

Tính tổng giá trị: O

**Đường đi giao Hàng:**

A white background with black numbers and text

Description automatically generated

A screenshot of a computer code

Description automatically generated

1. **Khởi tạo danh sách địa điểm và ma trận khoảng cách**: **O(n^2)**
   * Khởi tạo danh sách: **O(n)**
   * Khởi tạo ma trận: **O(n^2)**
2. **Tìm đường đi tối ưu (Hàm TimTuyenDuong)**:
   * **Vòng lặp đệ quy**:
     + Có n! cách sắp xếp các địa điểm.
   * **Tính toán khoảng cách cho mỗi sắp xếp**:
     + Mỗi sắp xếp có thể có đến n - 1 phép tính (khoảng cách giữa các địa điểm).
   * **Tổng độ phức tạp**:
     + **Tìm sắp xếp**: **O(n!)**
     + **Tính toán khoảng cách cho mỗi sắp xếp**: **O(n)**

**Tổng độ phức tạp**

* **Tìm đường đi tối ưu**: **O(n!)** (vì cần kiểm tra tất cả các hoán vị)