

1. Cấu trúc dữ liệu:

- `int m`; là số người thực hiện công việc.
- `int n`; là số công việc cần được phân công.
- `int time[MAX_N][MAX_N]`; là mảng 2 chiều lưu trữ thời gian để người thứ i thực hiện công việc thứ j .
- `int id[MAX_N][MAX_N]`; là mảng 2 chiều lưu trữ thông tin về id của m người thực hiện n công việc.
- `int machine[MAX_N]`; là mảng 1 chiều lưu trữ thông tin tổng thời gian thực hiện những công việc được phân công của người thứ i .
- `int solution[MAX_N]`; là mảng 1 chiều lưu trữ kết quả của bài toán, cho biết với công việc thứ j thì người nào thực hiện công việc đó.

2. Hàm sử dụng:

- Hàm `input(int time[][MAX_N], int id[][MAX_N], int &m, int &n)` dùng để nhập giá trị cho m , n , thời gian t_{ij} và thông tin id tương ứng của người thứ i thực hiện công việc thứ j .
- Hàm `arrange(int time[][MAX_N], int id[][MAX_N], int m, int n)`; dùng để sắp xếp toàn bộ giá trị `time[][]` theo nguyên tắc sau: Tương ứng với một cột, ta sắp xếp các giá trị trong mỗi cột đó theo thứ tự tăng dần.
- Hàm `int findMaxTimeOfMachines(int machine[], int m)`; dùng để cho biết người nào có tổng thời gian thực hiện công việc là lớn nhất.
- Hàm `findSolution(int time[][MAX_N], int id[][MAX_N], int machine[], int solution[], int m, int n)`; dùng để thực hiện tính toán để tìm ra kết quả của bài toán.
- Hàm `void output(int solution[], int machine[], int m, int n)`; dùng để xuất kết quả của bài toán lên màn hình.