

Các Bài Toán Cơ Bản

Bài 1

- Mã hóa bố cục lớp học thành các ô, trong đó ô tại hàng i và cột j là $(i-1) * 2 + (j-1)$.
- Xác định vị trí của Alice.
- Nếu vị trí này có thể trừ đi k , tìm ô có cùng mã với Alice ở phía trước.
- Nếu vị trí này có thể cộng thêm k , tìm ô có cùng mã với Alice ở phía sau.
- Nếu không có ô hợp lệ, in ra -1 .

Bài 2

- Định nghĩa một `struct Point` với các biến a , b , c và một hàm so sánh tùy chỉnh để so sánh các điểm.
- Sử dụng hàng đợi ưu tiên (priority queue) để sắp xếp các điểm theo yêu cầu.

Bài 3

- Viết hàm `merge` để trộn hai mảng bằng cách di chuyển con trỏ từ cả hai đầu.
- Nếu một mảng đã được duyệt hết, thêm tất cả các phần tử còn lại của mảng kia.

Bài 4

- Sử dụng hàng đợi ưu tiên để sắp xếp mảng và tìm MEX của mảng.

Bài 5

- Tương tự bài 2, nhưng hàm `Tpoint` lúc này chỉ có hai điểm y và z .

Bài 6

- Sử dụng hàng đợi ưu tiên để sắp xếp mảng và tính tổng của nó.
- Nếu tổng chia 3 dư 0, in ra chuỗi.

- Nếu tổng chia 3 dư 1, cố gắng xóa phần tử nhỏ nhất chia 3 dư 1. Nếu không thể, xóa hai phần tử chia 3 dư 2.
- Nếu tổng chia 3 dư 2, cố gắng xóa phần tử nhỏ nhất chia 3 dư 2. Nếu không thể, xóa hai phần tử chia 3 dư 1.

Bài 7

- Rời rạc hóa các phần tử.
- Duyệt qua các phần tử và in ra các phần tử phân biệt trong mảng.
- Nếu chưa đủ k , in ra các vị trí còn lại trong mảng.

Bài 8

- Lưu lại vị trí ban đầu của các phần tử trong mảng. Gọi v_u là vị trí ban đầu của phần tử thứ u .
- Sắp xếp lại mảng theo thứ tự tăng dần.
- Sau khi sắp xếp, mỗi phần tử sẽ có một vị trí mới. Sử dụng thông tin về vị trí ban đầu (v_u) để tạo các chu trình hoán đổi.
- Một chu trình được tạo khi ta bắt đầu từ một phần tử, di chuyển qua các vị trí ban đầu của các phần tử liên quan, cho đến khi quay lại phần tử ban đầu.
- Số lần hoán đổi cần thiết để đưa các phần tử trong một chu trình về đúng vị trí là số phần tử trong chu trình - 1.
- **Trường hợp các phần tử có giá trị giống nhau:**
 - Các phần tử giống nhau có thể tự đổi chỗ cho nhau mà không làm thay đổi số chu trình.
 - Trên chu trình tìm được, các phần tử giống nhau sẽ nằm liên tiếp nhau.
 - Do đó, chỉ cần không đếm các phần tử giống nhau liên tiếp trên chu trình là đủ.
- Tính tổng số lần hoán đổi cần thiết từ tất cả các chu trình để đưa mảng về đúng thứ tự.

Bài 9

- Xây dựng hàm `mergesort` như bình thường, thêm lệnh in kết quả ở cuối mỗi bước chia để trị.

Bài 10

- Thực hiện sắp xếp chèn (insertion sort). Sau mỗi lần hoán đổi hai phần tử, in ra mảng.

Bài 11

- Thực hiện sắp xếp nổi bọt (bubble sort). Sau mỗi lần hoán đổi hai phần tử, in ra mảng.

Bài 12

- Thực hiện sắp xếp chọn (selection sort). Sau mỗi lần hoán đổi hai phần tử, in ra mảng.

Các Bài Toán Nâng Cao

Bài 1

- Chuẩn bị để tính các hàm tổ hợp.
- Tính tổng các giá trị lớn nhất trong mảng trừ đi tổng các giá trị nhỏ nhất.
- Tổng các giá trị lớn nhất có thể tính bằng cách:
 - Duyệt các phần tử theo thứ tự giảm dần. Với phần tử i , có $n-i$ phần tử có thể chọn để i là lớn nhất.
 - Cộng $a[i] * C(k-1, n-i)$ vào kết quả.
- Tương tự để tính tổng các giá trị nhỏ nhất.

Bài 2

- Sử dụng `map<string>` để ánh xạ tài khoản với mật khẩu.
- Trong lúc truy vấn, nếu không tìm thấy, in ra `-1`.

Bài 3

- Tương tự bài 2.

Bài 4

- Sử dụng thuật toán KMP để tìm các vị trí khớp với chuỗi t trong mảng.
- Thuật toán KMP bao gồm hai bước:
 1. Xây dựng mảng lps (longest prefix which is also suffix) để tối ưu hóa việc so khớp.
 2. Duyệt qua chuỗi và sử dụng mảng lps để tìm tất cả các vị trí khớp.
- Độ phức tạp của thuật toán là $O(n + m)$ với n là độ dài chuỗi và m là độ dài mẫu.

Bài 5

- Tương tự bài 4.

Bài 6

- Chuẩn hóa các giá trị $a[i]$ thành một mảng lưu tần số xuất hiện của từng số.
- Chỉ quan tâm đến các số xuất hiện một lần và các số xuất hiện nhiều hơn một lần.
- Thuật toán:
 - Thêm k phần tử vào A , ưu tiên càng nhiều phần tử loại 2 càng tốt.
 - B sẽ chứa các phần tử còn lại.
 - Lần lượt chuyển các phần tử loại 2 từ A sang B nếu B chưa đủ, đảm bảo A không giảm và B tăng.
 - Nếu sau tất cả, B vẫn khác k , đáp án là NO ; ngược lại, là YES .

Bài 7

- Rời rạc hóa các phần tử trong mảng.
- Duyệt qua các phần tử và xác định trung vị của các đoạn con có độ dài k .
- Với mỗi đoạn con, sắp xếp các phần tử và lấy phần tử ở giữa làm trung vị.
- Do $\max(a[i]) \leq 200$, có thể tối ưu hóa bằng cách sử dụng mảng đếm tần số để tìm trung vị mà không cần sắp xếp.

Bài 8

- Lưu lại vị trí ban đầu của các phần tử trong mảng. Gọi v_u là vị trí ban đầu của phần tử thứ u .
- Sắp xếp lại mảng theo thứ tự tăng dần.
- Sau khi sắp xếp, mỗi phần tử sẽ có một vị trí mới. Sử dụng thông tin về vị trí ban đầu (v_u) để tạo các chu trình hoán đổi.
- Một chu trình được tạo khi ta bắt đầu từ một phần tử, di chuyển qua các vị trí ban đầu của các phần tử liên quan, cho đến khi quay lại phần tử ban đầu.
- Số lần hoán đổi cần thiết để đưa các phần tử trong một chu trình về đúng vị trí là số phần tử trong chu trình - 1.
- **Trường hợp các phần tử có giá trị giống nhau:**
 - Các phần tử giống nhau có thể tự đổi chỗ cho nhau mà không làm thay đổi số chu trình.

- Trên chu trình tìm được, các phần tử giống nhau sẽ nằm liên tiếp nhau.
- Do đó, chỉ cần không đếm các phần tử giống nhau liên tiếp trên chu trình là đủ.
- Tính tổng số lần hoán đổi cần thiết từ tất cả các chu trình để đưa mảng về đúng thứ tự.

Bài 9

- Gọi $x[i]$ là số thẻ thứ i sẽ được sử dụng trong mỗi lượt.
- Ta có: $x[1] + x[2] + \dots + x[n] = n$ với $(0 \leq x[i] \leq n)$.
- Sử dụng bài toán chia kẹo Euler để giải.
- Với N lớn và MOD nhỏ, sử dụng [định lý Lucas](#) để tính tổ hợp.