NUMBER

Lời giải

Ở bài giải này, bạn có thể sử dụng kĩ thuật quy hoạch động chữ số để giải quyết.
Gọi $dp[i][del][mod] = True/False$ có nghĩa là với i chữ số đầu tiên của số S, xóa đi del số và hiện tại đang có số dư là mod khi chia cho m thì có cách nào xử lí $n - i$ chữ số đằng sau sao cho thõa mãn xóa đi đúng k chữ số và chia hết cho m hay không.

Ban đầu ta có $dp[n][k][0] = True$ và tất cả trạng thái còn lại đều là $False$.

Để xây dựng được mảng dp đó thì ta có thể đệ quy có nhớ:

$dp[i][del][mod] = dp[i+1][del+1][mod]$ or $dp[i+1][del][(mod * 10 + digit(S,i))\%m]$
để mô tả quyết định xóa chữ số thứ i hoặc giữ chữ số thứ i lại.

Bây giờ để tìm được số lớn nhất thõa mãn điều kiện thì ta có nhận xét là: Vì số lượng chữ số cần tìm cố định là $n - k$ nên chữ số tính từ trái sang phải phải càng lớn càng tốt. Ta có cách làm tham lam sau:

- Giả sử đã chọn xong i chữ số từ trái sang, và chữ số thứ i là chữ số thứ L từ trái sang của số S, số lượng chữ số đã xóa sẽ là $del = L - i$ và có số dư hiện tại khi chia cho m là mod.
- Khi đó chữ số thứ $i + 1$ có thể được chọn là chữ số thứ j của số S trong khoảng $[L + 1, n - (n - k) + (i + 1)] = [L + 1, k + i + 1]$ sao cho $dp[j + 1][del + j - (i + 1)][(mod * 10 + digit(S, j))\%m] = True$. Nếu có nhiều j sao cho $digit(S, j)$ lớn nhất thì cách tối ưu sẽ chọn j nhỏ nhất có thể.

Cách giải cho số nhỏ nhất tương tự như trên nhưng cần xử lí trường hợp số 0 ở đầu.

Độ phức tạp: $O(n \times k \times m)$