

[Pages](#) / ... / [2.3 Dynamic Programming \(Thuật toán quy hoạch động\)](#)

## 2.3.1 Homework

Created by TUNG DUC NGUYEN tung2.nguyen, last modified on 2021/05/21

### Bài 1: House Robber (Medium)

<https://leetcode.com/problems/house-robber/>

Cho 1 mảng các số nguyên, yêu cầu lấy ra các số trong mảng đấy sao cho tổng là lớn nhất, với điều kiện không được lấy 2 phần tử cạnh nhau trong mảng.

Ví dụ: A = [1, 2, 3, 1]

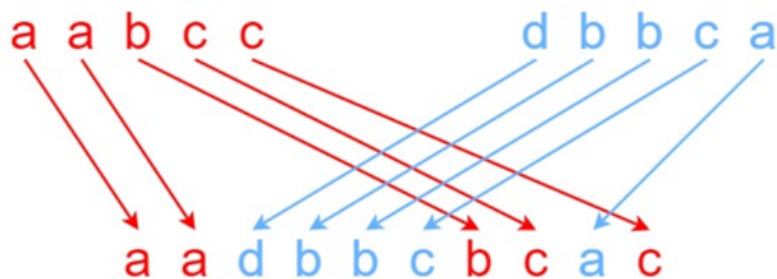
Output: 4. Lấy phần tử đầu tiên và thứ 3 của mảng sẽ cho tổng lớn nhất.

### Bài 2: Interleaving String (Medium)

<https://leetcode.com/problems/interleaving-string/>

Cho 3 chuỗi s1, s2, s3. Kiểm tra xem có thể tạo được chuỗi s3 từ việc phân tách và kết hợp 2 chuỗi s1 và s2.

Example 1:



Input: s1 = "aabcc", s2 = "dbbca", s3 = "aadbcbcbac"

Output: true

Giới hạn:  $0 \leq s1.length, s2.length \leq 100$

$0 \leq s3.length() \leq 200$

### Bài 3: Add One (Medium)

<https://codeforces.com/problemset/problem/1513/C>

Cho 1 số nguyên dương n, có m thắc tác thực hiện lên số nguyên dương này trong đó mỗi thao tác sẽ thay thế mọi chữ số d trong số n bằng d+1.

Ví dụ: n = 1912, m = 2

Thao tác 1: thay thay 1 bằng 2, 9 bằng 10, 2 bằng 3 tác có  $n = 21023$

Thao tác 2: thay 2 bằng 3, 1 bằng 2, 0 bằng 1, 2 bằng 3, 3 bằng 4 ta có  $n = 32134$

In ra số ký tự của số  $n$  thu được sau khi thực hiện  $m$  thao tác, số chữ số này có thể rất lớn, nên in ra theo phần dư của nó khi chia cho  $10^9 + 7$

Đầu vào:  $t$  testcase ( $1 \leq t \leq 2 \cdot 10^5$ ),  $n$  ( $1 \leq n \leq 10^9$ ),  $m$  ( $1 \leq m \leq 2 \cdot 10^5$ )

Đầu ra: mỗi testcase in ra số chữ số của số thu được

## Bài 4: String Reversal (Hard)

<https://codejam.lge.com/problem/19597>

Cho  $N$  strings, với mỗi string trong đây có thể chọn đảo ngược string hoặc giữ nguyên string, mỗi phép đảo ngược cho bit 1, không đảo ngược cho bit 0.

Tìm chuỗi bit có giá trị nhỏ nhất tương ứng để biến dãy  $N$  strings thành dãy có thứ tự tăng dần theo từ điển

VD:  $N = 3$ , and  $S[1] = "ABC"$ ,  $S[2] = "XC"$ , and  $S[3] = "DZ"$

Chuỗi bit: bit[1] = 0, bit[2] = 0; bit[3] = 0 ta có  $S[1] = "ABC"$ ,  $S[2] = "XC"$ , and  $S[3] = "DZ"$  thỏa mãn tăng dần theo từ điển

Input:

Dòng đầu tiên là số lượng test case  $T$  ( $1 \leq T \leq 50$ )

Với mỗi test case dòng đầu tiên là 1 số nguyên ứng với  $N$

$N$  dòng tiếp theo tương ứng với chuỗi  $s[i]$  ( $2 \leq \text{length } s[i] \leq 20$ )

Output:

In chuỗi bit có giá trị nhỏ nhất sao tương ứng với với chuỗi thao tác để biến  $n$  chuỗi  $s[i]$  tăng dần theo thứ tự từ điển

## Bài 5: Losing to Little Brother (Hard)

<https://codejam.lge.com/contest/problem/648/5>

Alice là chị gái của Albert. Hai chị em chơi trò chơi và vì Alice là chị gái của Albert nên Alice muốn nhường Albert. Alice sẽ chơi làm sao để thua Albert với số điểm nhỏ nhất.

Trò chơi như sau:

Trò chơi mà hai đứa trẻ này chơi sử dụng  $n$  thẻ, mỗi thẻ có một chữ cái tiếng Anh viết hoa (A-Z) trên đó.

Họ chơi trò chơi theo các quy tắc sau, và Alice biết chính xác Albert sẽ chơi như thế nào (vì anh ấy còn quá nhỏ!).

- Đầu tiên, hai bạn nhỏ sẽ xếp  $n$  thẻ xuống sàn từ trái sang phải theo thứ tự tùy ý.
- Lần lượt họ lấy một thẻ và người chơi chỉ có thể lấy thẻ ngoài cùng bên trái hoặc thẻ ngoài cùng bên phải.
- Albert đi trước, và sau đó họ thay phiên nhau.
  - Trong lượt của Albert, nếu Albert lấy bất kỳ thẻ nào có một trong sáu chữ cái trong "ALBERT", thì Albert được 2 điểm.
  - Trong lượt của Alice, nếu Alice lấy bất kỳ thẻ nào có một trong năm chữ cái trong "ALICE", thì Alice ghi được 1 điểm.
- Albert luôn lấy một thẻ theo mẫu sau:
  - Nếu cả thẻ ngoài cùng bên trái và thẻ ngoài cùng bên phải đều có thể cho anh ta điểm, Albert luôn lấy thẻ ngoài cùng bên trái.
  - Nếu không có thẻ nào trong hai thẻ có thể cho anh ta điểm, Albert luôn lấy thẻ ngoài cùng bên phải.
  - Nếu chỉ một trong hai thẻ có thể cho anh ta điểm, Albert luôn lấy thẻ có thể cho anh ta điểm.

Alice muốn thua Albert, nhưng cô ấy muốn giảm thiểu sự khác biệt về điểm số, (Albert's score - Alice's score).

Ví dụ, giả sử rằng  $n = 4$  và các thẻ trên sàn là "BCCB".

- Lượt 1: Albert sẽ lấy thẻ ngoài cùng bên trái "B", và ghi được 2 điểm. Các thẻ còn lại là "CCB".
- Lượt 2: Alice có thể lấy chữ "C" ngoài cùng bên trái hoặc chữ "B" ngoài cùng bên phải.
  - Nếu cô ấy lấy chữ "C" ngoài cùng bên trái: Cô ấy ghi được 1 điểm, và "CB" vẫn còn. Albert lấy chữ "B" ngoài cùng bên phải và Alice lấy chữ "C" cuối cùng. Điểm của Albert là 4 và điểm của Alice là 2 (hiệu số là 2).
  - Nếu cô ấy lấy chữ "B" ngoài cùng bên phải: Cô ấy không ghi được điểm nào và "CC" vẫn còn. Albert lấy chữ "C" ngoài cùng bên phải, và Alice lấy chữ "C" cuối cùng. Điểm của Albert là 2 và điểm của Alice là 1 (hiệu số là 1).
- Trong ví dụ này, nếu Alice cố gắng hết sức để thua Albert với cách biệt điểm nhỏ nhất, thì cô ấy chỉ có thể thua với cách biệt 1 điểm.

Trong một ví dụ khác, giả sử rằng  $n = 4$  và các thẻ trên sàn là "CLCD".

- Lượt 1: Albert sẽ lấy thẻ ngoài cùng bên phải "D", và ghi 0 điểm. Các thẻ còn lại là "CLC".
- Lượt 2: Alice có thể lấy chữ "C" ngoài cùng bên trái hoặc chữ "C" ngoài cùng bên phải.
  - Nếu cô ấy lấy chữ "C" ngoài cùng bên trái: Cô ấy ghi được 1 điểm và "LC" vẫn còn. Albert lấy chữ "L" ngoài cùng bên trái (và điểm 2) và Alice lấy chữ "C" cuối cùng. Đó là một trận hòa khi cả hai người chơi ghi được 2 điểm.
  - Nếu cô ấy lấy chữ "C" ngoài cùng bên phải: Cô ấy ghi được 1 điểm, và "CL" vẫn còn. Albert lấy chữ "L" ngoài cùng bên phải (và điểm 2), và Alice lấy chữ "C" cuối cùng. Đó là một trận hòa khi cả hai người chơi ghi được 2 điểm.
- Trong ví dụ này, nếu Albert lần lượt lấy chữ "C" ngoài cùng bên trái, thì Alice có thể lấy chữ "D" ngoài cùng bên phải, tiếp theo là Albert lấy chữ "D" ngoài cùng bên phải và Alice lấy chữ "L" cuối cùng, sẽ có dẫn đến việc Alice thua 1 điểm. Tuy nhiên, Albert luôn lấy thẻ ngoài cùng bên phải khi không có thẻ nào cho anh ta điểm, và do đó Alice không thể thua Albert trong ví dụ này.

Đưa ra  $n$  thẻ trên sàn (từ trái sang phải), tính chênh lệch điểm số nhỏ nhất mà Alice có thể thua Albert.

★★★★: Problems and explanations are contributed by @HUNG TRONG HO [hung.ho](#) @NAM TIEN NGUYEN [nam4.nguyen](#) @TIEN THANH NGUYEN [tien.nguyen](#)

No labels