P1 : UVA 524 – Prime Ring Problem (Brute force + pruning )

<https://uva.onlinejudge.org/index.php?option=onlinejudge&page=show_problem&problem=465>

P3 : CF 492C - Vanya and Exams (Greedy)

<http://codeforces.com/problemset/problem/492/C>

sort theo chiều giảm của số essay cần viết

P4,5: Easy

P6 : TIMUS – 1761 – unsolved (can solve by Hash)

<http://acm.timus.ru/problem.aspx?space=1&num=1761>

P7 : CF 279B – Books (Two Pointers or Chặt nhị phân)

<http://codeforces.com/problemset/problem/279/B>

Chặt nhị phân :

sao khi cộng dồn tổng , cho I từ 0 <= n

Ứng với mỗi s[i] -> tìm s[j] sao cho s[j] – s[i] <= t bằng cách chặt nhị phân

res = max (res, j-i);

Chặc nhị phân đáp án(Binary Search Anwers)

Để ý ta thấy, nếu tồn tại cách chọn một dãy con gồm m phần thử thỏa mảng thì sẽ tồn tại cách chọn một đoạn < m phần thử.

Lợi dụng diểm này ta chặc nhị phân như sao.

Với đoạn [l, r], gọi m = (l + r) / 2. Nếu có cách chọn m quyển sách sao cho tổng <= t. thì ta ghi nhận lại m, Thu gọn l = m + 1. ngược lại ta thu gọn r = m - 1(vì từ m - > r chắc chắc ko có giá trị nào thỏa mảng). Làm cho tới khi l > r.(Không thể phân đôi được nửa). Chọn giá trị tốt nhất trong các giá trị đả ghi nhận.

P8 : TIMUS 1225 – FLAGS

<http://acm.timus.ru/problem.aspx?space=1&num=1225>

Dp dễ

Có mảng 2 chiều f[3][46];

Với 3 là 3 trạng thái của màu, 46 là số lượng stripes

Ta có f dc tính như sau (theo yêu cầu của đề) :

f[0][i] = f[1][i-1]/2 + f[2][i-1];

f[1][i] = f[0][i-1] + f[2][i-1];

f[2][i] = f[0][i-1] + f[1][i-1]/2;

cuối cùng ta sum f[0][n-1]+f[2][n-1] lại là ra kết quả cần tìm

P9 : TJU 3639 – Bag Problem (Phân đôi + Chặt nhị phân)

<http://acm.tju.edu.cn/toj/showp3539.html>

Sau khi phân đổi thành 2^20 mỗi mảng, ta sinh các trường hợp có thể xảy ra vào từng mảng, điều kiện sinh là sum <= m và số lượng vật dùng <= n

Sau đó ta sẽ tổ hợp các phần tử của A và B với nhau, ứng với A ta sẽ dùng chặt nhị phân để tìm B sao cho (sumA + sumB <= m và số phần tử A + số phần tử B <= n ) , B đã được sort theo sum để break khi đến sumB cần thiết.

P10 : Stack đơn giản