1. **BOWLING**

* SUB 1: Duyệt nhị phân
* SUB 2: Dp[i] xét đến quả thứ i ném quả i thì được nhiều nhất bao nhiêu điểm. Công thức: dp[i] = min(dp[j] + cost) ( j = 1 .. i-1 ).
* SUB 3: Mô hình bài toán về việc chọn các đoạn con liên tiếp có độ dài >=

2 \* r + 1. Chú ý ở đây ở vị trí 1và n ta có thể chọn được các đoạn con có độ dài >= r + 1 ( Việc này ta có thể thêm r số 0 ở đầu và ở cuối ). Ta có mô hình Dp như sau dp[i][0/1] xét đến số thứ i có chọn số thứ i để nối tiếp đoạn >= 2 \* r + 1 hoặc là 0 chọn. Công thức:

dp[i][0] = max(dp[i-1][0], dp[i-1][1]);

dp[i][1] = dp[i-1][1] + a[i];

dp[i][1] = max(dp[i][1], dp[i-2\*r-1] + S[i-2\*r+1… i]) ( nếu i >= 2 \* r + 1).

1. **SCHOOL**

* Mô hình đường đi đến trường của Hồng chỉ có thể là đi tàu đến 1 điểm nào đó rồi sử dụng các đường tắt để đi đến trường. Tức là sẽ là min(f[i] + g[i]).

Trong đó f[i] là chi phí dùng tàu đi đến điểm i, g[i] là sử dụng đường tắt nhỏ nhất đi từ điểm i đến trường.

* Ta tính các mảng f, g như sau. Mảng g ta có thể dùng BFS, mảng f chính là đường đi theo tuần tự. Ở mỗi truy vấn thì ta cập nhật bằng segment tree cho giá trị của fi mới sau khi bị đổi chỗ.

1. **LIGHT**

* SUB 2: Ta nhận thấy việc bật đèn để chuyển màu thì chỉ liên quan đến số lần bật chẵn hay lẻ từ đó ta có được ở mỗi đèn i chọn bật hay không bật. Qui ước 0 là xanh, 1 là đỏ thì mỗi đèn ta có phương trình:

Số hiệu đèn hiện tại (1 đỏ/ 0 xanh) + 1 \* (xi + 1 \* (x các đèn có cạnh nối với i) % 2 = 0

Đến đây ta dùng khử Gauss để giải quyết bài toán.

* SUB 3: Việc dùng khử Gauss từng hàng ta có thể làm trong phép XOR. Và việc trả lời T truy vấn thì ta có thể sắp cả T truy vấn thành T cột vào bảng khử Gauss. Từ đó ta có thể tối ưu bài toán từ sub2.