**A – Prime**

Sub 1 + 2:

Với mỗi số nguyên từ 2 đến , kiểm tra xem có phải là ước của không.

Độ phức tạp:

Sub 3:

Sử dụng Miller-Rabin test, kiểm tra 1 một số có thể là số nguyên tố với tỉ lệ sai số là (k là số lần lặp test)

Độ phức tạp:

**B – Polish Notation**

Sử dụng stack để lưu lại các số trong quá trình tính toán, khi gặp 1 dấu phép tính, thực hiện phép tính với 2 số ở đỉnh của stack.

**C – Fibonacci Rectangle**

Với hình chữ nhật có kích thước , ta cần cắt một hình vuông có cạnh sao cho hình vuông được cắt ra không chứa tọa độ ta cần cho hình vuông .

Gọi tọa độ trái cùng và phải cùng của hình chữ nhật hiện tại lần lượt là và , nếu hoặc thì không tồn tại cách cắt để cắt được hình chữ nhật ở tọa độ . Liên tục cắt hình chữ nhật cho đến khi còn lại 1 hình vuông kích thước hay .

Độ phức tạp: cho mỗi test case

**D – Transmit**

Sub 1: Dựng đồ thị với đỉnh và tối đa cạnh. Sau đó thực hiện truy vấn qua kiểm tra vùng liên thông

Độ phức tạp:

Sub 2: Một số nguyên sẽ có tối đa là ước nguyên tố. Sử dụng tính chất này, ta có thể tạo thêm các đỉnh ảo là các ước nguyên tố có trong đỉnh ban đầu, tối đa là đỉnh ảo. Từ N đỉnh ban đầu, nối cạnh đến với các đỉnh ảo là ước nguyên tố của nó. Mỗi truy vấn giờ sẽ yêu cầu tìm xem và có cùng một vùng liên thông không.Điều này có thể khả thi trong thời gian cho phép bằng cách cài đặt Cây khung nhỏ nhất bằng Disjoint-set Union để chuẩn bị cho truy vấn

Độ phức tạp:

**E – Explosion**

Ta nhận thấy rằng, sát thương của Explosion càng lớn, thì ta sẽ càng phải thi triển Burn ít hơn. Vì thế ta sẽ có kết quả của bài toán là

Để có thể hạ toàn bộ quái bằng chiêu Explosion, khi ta thi triển Explosion vào con quái thứ thì mảng H sẽ tăng nghiêm ngặt từ 1 đến i và giảm nghiêm ngặt từ i đến n (sau khi lược bỏ các quái có 0 HP ở đầu và cuối dãy). Ta có thể chắc chắn rằng sử dụng Explosion với H(i) MP là tối ưu trong mọi trường hợp.

Sub 1: Với mỗi con quái , ta có thể tính lượng sát thương tối đa khi sử dụng Explosion lên nó sau khi đã dùng Burn để giảm máu các con quái xung quanh một cách tối ưu:

Note: dmg = H(i)

Tập trung vào vế trái của mảng H. Khi gây sát thương vào con quái thứ thì con quái thứ sẽ nhận sát thương. Khi đó lượng sát thương mà nó có thể lan tới con quái tiếp theo là:

Việc tính toán này sẽ dừng lại khi lượng sát thương lan ra bằng 0

Từ đó, ta có thể tính được sát thương tối đa của Explosion.

Độ phức tạp: 2)

Sub 2:

Tiếp tục với vế trái của mảng H, ta có thể tổng quát hóa sát thương gây ra trên con quái thứ khi thực hiện Explosion trên con quái thứ là:

Công thức trên sẽ đúng cho đến khi (hết sát thương), (hết quái), hoặc . Hai trường hợp đầu có thể được kiểm tra trong thời gian . Với trường hợp thứ 3, khi ta hạ con quái thứ bằng sát thương lan của Explosion, nó sẽ gây sát thương lan mà không phụ thuộc vào , đưa bài toán trờ thành “Tính sát thương lan nếu chúng ta bắt đầu từ quái thứ ”.

Từ ý trên, ta có ý tưởng quy hoạch động như sau: gọi là sát thương lan cao nhất khi sử dụng Explosion lên con quái thứ . Ta cần tìm con quái gần nhất mà . Trên đoạn , ta biết rằng sát thương gây ra trên các con quái theo quy luật: cho con thứ , cho con thứ , …, cho con thứ . Tổng lại, lượng sát thương được gây ra là .

Về việc tìm gần nhất sao cho hay . Bài toán tương đương với tìm gần nhất sao cho với và có thể xử lý với một stack.

Duyệt và duy trì một stack của các trước đó. Khi ta cần tìm thỏa mãn , ta chỉ cần kiểm tra đỉnh stack, nếu thì ta đã tìm được thỏa mãn, ngược lại ta pop đỉnh của stack và tiếp tục kiểm tra.

Do mỗi phần tử được push vào stack và pop khỏi stack 1 lần nên thuật toán có độ phức tạp

Thực hiện tính toán tương tự với mảng ngược lại so với mảng .

Đáp án cho bài toán là:

Độ phức tạp: