|  |  |
| --- | --- |
| SỞ GIÁO DỤC VÀ ĐÀO TẠO | **KỲ THI OLYMPIC TRUYỀN THỐNG 30 THÁNG 4** |
| TỈNH BÀ RỊA - VŨNG TÀU | **LẦN THỨ XXVIII – NĂM 2024** |
| **TRƯỜNG THPT CHUYÊN** | **ĐÁP ÁN CHÍNH THỨC** |
| **LÊ QUÝ ĐÔN** |  |
|  | MÔN THI: **TIN HỌC** - KHỐI: **10** |
| (Đáp án này gồm có **04** trang) |

**Câu 1.** *(6 điểm)* **BĂNG RÔN OLYMPIC**

Có tất cả 20 test, mỗi test được 0.3 điểm với phân bố như ràng buộc ở đề bài.

***Subtask 1***: Có 25% số test ứng với 25% số điểm của bài thỏa mãn: 

Duyệt vét cạn hai lần theo cặp chỉ số  sao cho .

Ứng với mỗi cặp chỉ số, sử dụng thêm một vòng lặp nữa để đếm các ký tự  trong chuỗi con  của băng rôn đã cho xem chuỗi con có thỏa mãn điều kiện “đẹp” hay không. Mỗi lần được thêm một chuỗi con thỏa mãn thì tăng biến đếm kết quả lên 1 đơn vị.

Độ phức tạp: 

***Subtask 2***: Có 25% số test ứng với 25% số điểm của bài thỏa mãn: 

Để cải tiến hơn so với cách trên, ta tạo trước ba mảng cộng dồn  đếm số lượng chữ cái  trong đoạn từ  đến  Để đếm số chữ cái  trong chuỗi con , ta chỉ cần tính giá trị  là được; tương tự với các chữ cái  sẽ lần lượt là  và .

Như vậy, ở mỗi lần thực hiện hai vòng lặp, ta chỉ tốn chi phí  để kiểm tra xem chuỗi con tương ứng có thỏa mãn tính chất đẹp hay không.

Độ phức tạp: 

***Subtask 3***: Có 50% số test ứng với 50% số điểm của bài thỏa mãn: 

Ta có nhận xét rằng một chuỗi con có độ dài ít nhất là  thì theo nguyên lý Dirichlet, sẽ có một ký tự xuất hiện ít nhất  lần trở lên nên thỏa mãn ràng buộc về chuỗi đẹp. Như vậy, ứng với mỗi vị trí , ta sẽ kiểm tra trực tiếp xem có bao nhiêu chuỗi con ứng với  là thỏa mãn, còn với  trở đi thì chắc chắn là đẹp nên sẽ lấy hết các chuỗi con này.

Độ phức tạp: 

*Cách khác:* Ngoài ra, ta cũng có thể khai thác tiếp mảng cộng dồn ở subtask 2 bằng việc thực hiện tìm kiếm nhị phân xem ứng với mỗi  thì chỉ số  đầu tiên nào thỏa mãn một trong các ràng buộc  là được.

Độ phức tạp: 

**Câu 2.** *(7 điểm)* **ĐƯỜNG ỐNG THOÁT NƯỚC**

Có tất cả 28 test, mỗi test được 0.25 điểm với phân bố như ràng buộc ở đề bài.

***Subtask 1***: Có 25% số test ứng với 25% số điểm của bài thỏa mãn: 

Ta coi hệ thống đường ống thoát nước này là một graph đơn vô hướng  có tất cả  đỉnh và  cạnh.

Yêu cầu của bài toán là: chọn ra không quá  đỉnh có cùng bậc sao cho số lượng đỉnh của  có đường đi đến đỉnh nào đó của  đỉnh được chọn là nhiều nhất, cho biết số lượng nhiều nhất này.

Trước hết, ta thực hiện duyệt DFS qua các đỉnh để chia graph thành các thành phần liên thông trong độ phức tạp , đánh dấu mỗi đỉnh với chỉ số của thành phần liên thông tương ứng. Hơn nữa, ta cũng có thể cài đặt thích hợp để kiểm tra được mỗi thành phần liên thông sẽ bao gồm các đỉnh nào và bậc của chúng cụ thể là bao nhiêu. Nhận xét rằng nếu chọn từ một thành phần ra một đỉnh nào đó thì ta có thể lấy được hết số lượng đỉnh của thành phần liên thông đó cộng vào kết quả cần tìm.

Ở đây chỉ cần lấy ra  đỉnh nên đơn giản là chọn trong các thành phần liên thông ở trên ra thành phần có kích thước lớn nhất là được.

Độ phức tạp: 

*Ghi chú:* có thể dùng thuật toán DSU để tìm số thành phần liên thông của graph  đã cho với độ phức tạp 

***Subtask 2***: Có 25% số test ứng với 25% số điểm của bài thỏa mãn: 

Với giá trị  tùy ý, vấn đề là làm sao chọn được các đỉnh có cùng bậc và tổng kích thước các thành phần liên thông chứa chúng là lớn nhất. Rõ ràng các bậc của các đỉnh có thể dao động từ  đến . Ta thực hiện duyệt qua các bậc  đó, ứng với mỗi bậc thì duyệt qua tiếp các thành phần liên thông xem có những thành phần nào chứa đỉnh có bậc  thì lưu thông tin kích thước của nó lại thành một danh sách 

Ta sắp xếp danh sách  này lại và chọn ra  giá trị lớn nhất trong đó rồi lấy tổng của chúng, đây là kết quả lớn nhất ứng với việc chọn ra các đỉnh bậc  Đáp số của bài toán sẽ là giá trị lớn nhất trong các kết quả trên khi kiểm tra hết các giá trị 

Độ phức tạp: 

***Subtask 3***: Có 50% số test ứng với 50% số điểm của bài thỏa mãn các ràng buộc ban đầu của đề.

Ta thực hiện tương tự trên nhưng có cải tiến bước duyệt kết hợp với cách lưu trữ cẩn thận hơn. Với mỗi đỉnh, ta sẽ cài đặt thích hợp để lưu chỉ số của thành phần liên thông chứa nó. Duyệt qua các đỉnh , ta tính được bậc của , giả sử là  Trước đó, ta tạo một set để lưu lại chỉ số của các thành phần liên thông mà trong đó có chứa một đỉnh có bậc bằng , đặt là  và có thể xây dựng được các set này thông qua mỗi lần duyệt trên (ta dùng set để tránh việc hai đỉnh cùng một thành phần bị tính lặp lại).

Ứng với thông tin trong mỗi , ta sẽ đổi chỉ số của các thành phần liên thông thành kích thước của nó và lưu thành một mảng, sắp xếp mảng đó lại để tính tổng của  số lớn nhất là được, đây là kết quả lớn nhất ứng với việc chọn ra các đỉnh bậc  Đáp số của bài toán sẽ là giá trị lớn nhất trong các kết quả trên khi kiểm tra hết các giá trị 

Độ phức tạp: 

**Câu 3.** *(7 điểm)* **HỌC MÁY**

Có tất cả 50 test, mỗi test được 0.14 điểm với phân bố như ràng buộc ở đề bài.

***Subtask 1***: Có 20% số test ứng với 20% số điểm của bài thỏa mãn  tập tin đều có kích thước không vượt quá 

Với kích thước tập tin  nhỏ, ta có thể dùng quy hoạch động. Gọi  là tổng nhỏ nhất cần tìm ứng với số  thì  Với mỗi số  ta kiểm tra xem có thể biểu diễn nó ở dạng  hay không, và nếu được thì số mũ  nhỏ nhất là bao nhiêu.

Ứng với mỗi  ta duyệt  và tiếp tục duyệt  để xem có đẳng thức  hay không (chú ý rằng khi  thì ) nên tốn chi phí .

Từ đó, ta sẽ có mảng  với , trong đó

*  nếu  không thể biểu diễn thành dạng lũy thừa như yêu cầu.
*  với  là số mũ nhỏ nhất để có biểu diễn .

Với mỗi  ta sẽ duyệt tất cả số  để cập nhật mảng  như sau :

 với .

Sau khi tính được xong mảng  ta trả lời được đồng thời  truy vấn dễ dàng. Kết quả cần tìm cho tập tin kích thước  sẽ là 

Độ phức tạp 

***Subtask 2***: Có 40% số test ứng với 40% số điểm của bài thỏa mãn:  với  là tổng kích thước các tập tin.

Ta cải tiến bước tính  ở trên như sau: nhận xét rằng khi  thì số mũ  trong biểu diễn  sẽ thỏa mãn  nên ta kiểm tra xem  có phải là lũy thừa đúng bậc  hay không. Điều này có thể thực hiện được nhờ thư viện pow có sẵn, chi phí tính toán là Ta lưu hết các giá trị  ứng với số mũ  này lại thành một mảng  để duyệt trên đó, có thể kiểm tra được kích thước mảng này không quá lớn, nếu đặt nó là  thì .

Để cập nhật mảng  vẫn dùng công thức trên nhưng thay vì duyệt tất cả số  thì ta chỉ cần duyệt trên mảng  với các số không vượt quá  là được.

Độ phức tạp 

***Subtask 3***: Có 40% số test ứng với 40% số điểm của bài thỏa mãn:  với  là tổng kích thước các tập tin.

Với giá trị  lớn, ta không thể duyệt vét cạn  để cập nhật mảng  mà cần khảo sát sâu hơn tính chất của các số  trong tổng



Ý tưởng chính ở đây là dựa theo tính chất đã được gợi ý “ngầm” trong phần mô tả của đề bài. Ta thấy rằng chắc chắn số  sẽ luôn viết được thành tổng của không quá  bình phương nên đáp số của bài toán sẽ bị chặn trên bởi  đó là một nhận xét quan trọng để có thể chia trường hợp.

Chú ý thêm rằng ta không cần xét lũy thừa bậc chẵn lớn hơn  vì luôn có thể quy về bậc  bởi công thức  nên điểm khó chính là ở bậc lẻ. Ta kiểm tra lần lượt tính chất của  theo thứ tự sau (nếu thỏa mãn điều kiện ở bước nào thì dừng ngay ở bước đó):

(1) Nếu  là số chính phương, kết quả là  Độ phức tạp: 

(2) Nếu  là lập phương đúng, kết quả là . Độ phức tạp: 

(3) Nếu  có dạng , kết quả là 4. Để kiểm tra, ta vét cạn các số  rồi kiểm tra  có chính phương hay không. Độ phức tạp: 

(4) Nếu  có dạng  ta cũng vét cạn tương tự trên, kết quả là 5. Ngoài ra, nếu  có dạng  thì cũng có kết quả bằng 5. Độ phức tạp chung cho hai trường hợp là 

(5) Nếu  có dạng  thì kết quả là 6, ta vét cạn với độ phức tạp . Thêm vào đó, nếu  có dạng  thì cũng có kết quả là , vấn đề là làm sao kiểm tra được? Đây là bước khó vì về lý thuyết, nếu thực hiện hai vòng lặp for lồng nhau thì độ phức tạp sẽ là .

Ta xử lý khéo léo như sau: lúc này,  đã không là bình phương hoặc tổng  bình phương nữa, khi đó nó có thể biểu diễn thành tổng của  hoặc  bình phương. Ta dùng tính chất đã nêu ở đầu bài để kiểm tra xem  có phải ở dạng  hay không trong chi phí .

Nếu nó không có dạng này thì nó sẽ biểu diễn được thành tổng ba bình phương như mong muốn. Độ phức tạp chung cho hai trường hợp là 

(6) Nếu  có các dạng , hoặc  hoặc  thì đều có kết quả là 7. Độ phức tạp lớn nhất là  ứng với hai vòng lặp duyệt cho căn bậc hai và căn bậc ba lồng nhau, vẫn đủ để chạy trong thời gian cho phép với  (chú ý rằng trên thực tế thì sẽ không cần đến  vì vòng lặp thứ hai sẽ có có chi phí giảm dần theo vòng lặp đầu).

(7) Nếu  không có các dạng trên thì đáp số là .

Độ phức tạp: 

**--------- HẾT ---------**