|  |  |
| --- | --- |
| **SỞ GIÁO DỤC VÀ ĐÀO TẠO**  **NAM ĐỊNH** | **ĐÁP ÁN VÀ HƯỚNG DẪN CHẤM**  **ĐỀ THI CHỌN HỌC SINH GIỎI TỈNH**  **NĂM HỌC 2023 - 2024**  Môn: **TIN HỌC** – Lớp: **12 THPT** |

HƯỚNG DẪN CHẤM

**\* Chấm các bài thi bằng các test, có phân chia mức độ.**

- Chấm các bài làm bằng bộ test, đúng test nào được điểm tương ứng của test đó. Các test đánh giá mức độ tối ưu của thuật toán.

- Nếu bài nào 0 điểm thì kiểm tra tệp chương trình, nếu chỉ là lỗi do đặt sai tên tệp chương trình thì sửa lại cho đúng và chấm lại; trừ 10% số điểm của bài đó đạt được. (Lỗi này có thể do thí sinh không để ý đến phần đặc trưng của tệp)

GIẢI THUẬT THAM KHẢO:

**Bài 1. (4 điểm) Thú cưng**

- Số thức ăn tổng hợp dành thêm cho chó: x 🡨 n- a; nếu x<0 thì x🡨0

- Số thức ăn tổng hợp dành thêm cho mèo: y 🡨 m- b; nếu y<0 thì y🡨0

- Cần mua thêm thức ăn tổng hợp: t🡨; nếu t<0 thì t🡨0

**Bài 2. (4,0 điểm) Số chẵn**

* Do các số có tổng các chữ số chẵn/ lẻ tương đương trong phạm vi từ 0 đến M-1

(M=N-N mod 10)

* Số các số thỏa mãn yêu cầu < M : k 🡨 M/2 -1 (số 0 bắt đầu)
* Chỉ cần kiểm tra thêm các số còn lại: từ M đến N

**Bài 3. (4,0 điểm) Đèn nhấp nháy**

**-** s🡨s+’.’ **;**  t🡨1; k🡨0

**- Với mỗi i nhận giá trị từ 1 đến n**

+ nếu si=si+1 thì t🡨t+1

Không thì gán k🡨k+t div 2 ; t🡨1

- In ra kết quả bài toán là k.

**Bài 4. (4,0 điểm) Dây đủ màu**

\* Các test tương ứng với 50% số điểm: có ***N*** và ***M*** không quá 103

* Duyệt các phần tử của dãy: Với mỗi phần tử thứ i xét cặp (i, j) thỏa mãn 1 ≤ i ≤ j ≤ N, nếu A[j] chưa xuất hiện thì đánh dấu A[j] và tăng biến *dem* cho đến khi bằng M
* Nếu *dem*=M thì cập nhật giá trị Min( j-i+1)

\* Các test tương ứng với 50% số điểm: có ***N*** và ***M*** không quá 107

* Gọi B[k] số lần xuất hiện mã **k** trong đoạn [i, j]; Ban đầu các giá trị B bằng 0
* Xét [i,j] dựa theo [i, j-1]: tăng số lần xuất hiện mã aj;

nếu aj=1 (vừa thêm mã mới) thì tăng ***dem***. Khi đủ mã: cập nhật giá trị MIN

* Xét [i, j] dựa theo [i-1, j]: giảm số lần xuất hiện mã ai;

nếu ai =0 (vừa bớt mã cũ) thì giảm ***dem***.

Không thì (vẫn đủ mã): cập nhật giá trị MIN

*Tham khảo*:

* i🡨 1; j🡨0 ; dem🡨0; D🡨0
* ***lặp khi j<=N***

+ b1: j 🡨 j+1, B[a[j] ] 🡨B[ a[j] ] +1

+ b2: nếu B[a[j] ] = 1 thì dem🡨dem+1

+ b3: lặp nếu dem =M

. D🡨min(D, j-i+1);

. i🡨i+1; B[a[j] ] 🡨 B[a[i] ]-1

. nếu B[a[j] ] =0 thì dem🡨dem-1

* In ra D

**Bài 5. (4,0 điểm) Chọn quà**

\* Các test tương ứng với 50% số điểm: có ***N*** ≤ 10; các số ***Gi*** đều là số nguyên tố.

Xét cấu hình nhị phân (chọn hay không chọn quà i). Cập nhật giá trị tốt nhất.

\* Các test tương ứng với 25% số điểm: có ***N*** ≤ 100 và số bộ dữ liệu là 1.

(Sử dụng phương pháp quy hoạch động)

* Fj – số lần phân chia ít nhất để đạt được giá trị j (ban đầu = 101)
* Hj – Hộp quà cuối cùng dùng để đạt được giá trị quà j (ban đầu = 0)
* Thử đến hộp quà i, với các cách phân chia

+ Với **gi**

++ kiểm tra j từ S 🡪**Gi**

+++ nếu ( Fj-gi <Fj ) thì Fj = Fj-gi ; Hj=i

+ Với gx = ( gi/2, gi/3, gi/5,…)

++ kiểm tra j từ S 🡪 **Gi**

+++ nếu ( Fj-gx+1 <Fj ) và **Hj<>i** thì Fj = Fj-gx+1 ; Hj=i

- Kết quả: Fk với (Fk <101 và k lớn nhất và k ≤ S)

\* Các test tương ứng với 25% số điểm: có ***N*** ≤ 100, nhiều bộ dữ liệu.

+ Tương tự phần ràng buộc 2, cần tối ưu chạy lùi từ min(S, tổng quà (1..i) )

+ Do Gi trong phạm vi 1000, nên lưu và xét chia với 11 số nguyên tố đầu tiên là đủ.

Hết