

ĐỀ THI CHÍNH THỨC

Môn: **TIN HỌC**

Thời gian: **300** phút (*không kể thời gian giao đề*)

Ngày thi: **27/4/2022**

Đề thi gồm 05 trang, 03 bài

TỔNG QUAN ĐỀ THI

Bài	Tên bài	Tên file chương trình
4	Numino	numino.cpp
5	Nén dữ liệu	apzip.cpp
6	Phần thưởng	bonus.cpp

Lập trình giải các bài toán sau đây:

Bài 4. (100 điểm) Numino

Lê rất thích trò chơi domino từ khi còn bé. Lấy cảm hứng từ trò chơi domino, gần đây Lê sáng tạo ra trò chơi mới có tên là numino gồm n quân. Mỗi quân numino là một hình chữ nhật được chia làm hai nửa trên cùng một mặt, trên mỗi nửa có ghi một số nguyên không âm. Các con số trên các nửa của các quân numino có thể khác nhau, nhưng tổng hai số trên hai nửa của mỗi quân numino đều bằng một giá trị cố định s .

Lê đặt n quân numino lên bàn, xếp chúng thành một dãy ngang, mỗi quân đều xếp theo chiều dọc vào dãy và đánh số các quân lần lượt từ 1 đến n theo thứ tự từ trái qua phải. Như thế, mỗi quân numino trong dãy sẽ có một nửa ở trên và một nửa ở dưới. Ban đầu, ở quân numino thứ i , nửa trên ghi số a_i và nửa dưới ghi số $(s - a_i)$. Lê lần lượt thực hiện q thao tác với các quân numino của mình, mỗi thao tác thuộc một trong ba loại sau:

- (1) $> \ell r b$: Lê xét tất cả các quân numino có vị trí từ ℓ đến r , nếu con số ở nửa trên của một quân numino nào đó có giá trị lớn hơn b thì Lê sẽ đảo hai nửa bằng cách xoay ngược quân numino này.
- (2) $< \ell r b$: Lê xét tất cả các quân numino có vị trí từ ℓ đến r , nếu con số ở nửa trên của một quân numino nào đó có giá trị nhỏ hơn b thì Lê sẽ đảo hai nửa bằng cách xoay ngược quân numino này.
- (3) $? p$: Lê ghi ra vở con số ở nửa trên của quân numino thứ p tại thời điểm đang xét dãy.

Khi Lê xoay ngược một quân numino, hai nửa của quân numino sẽ đổi chỗ cho nhau. Nói cách khác, giả sử một quân numino có số x ở nửa trên và số y ở nửa dưới, thì sau khi xoay ngược quân numino này, con số ở nửa trên sẽ là y và con số ở nửa dưới là x . Lưu ý rằng, với mỗi thao tác loại 1 và 2, Lê chỉ có thể xoay ngược các quân numino chứ không thay đổi thứ tự vị trí của chúng ở dãy xuất phát.

Yêu cầu: Hãy viết chương trình giúp Lê ghi lại con số ở nửa trên các quân numino trong các thao tác 3.

Dữ liệu: Vào từ thiết bị vào chuẩn:

- Dòng đầu chứa ba số nguyên dương n, s, q ($n, q \leq 5 \times 10^5, s \leq 10^7$) lần lượt là số quân numino, tổng giá trị hai số trên mỗi quân numino và số thao tác Lê sẽ thực hiện.
- Dòng thứ hai chứa n số nguyên không âm a_1, a_2, \dots, a_n ($a_i \leq s$), với a_i là số ở nửa trên của quân numino thứ i ($1 \leq i \leq n$).
- Mỗi dòng trong số q dòng cuối cùng mô tả một thao tác thuộc một trong ba loại ở trên:
 - (1) $> \ell r b$: Loại thứ nhất gồm một kí tự $>$, tiếp theo là dấu cách và 3 số nguyên dương $\ell r b$;
 - (2) $< \ell r b$: Loại thứ hai gồm một kí tự $<$, tiếp theo là dấu cách và 3 số nguyên dương $\ell r b$;
 - (3) $? p$: Loại thứ ba gồm một kí tự $?$, tiếp theo là dấu cách và 1 số nguyên dương p ;
 ($1 \leq \ell \leq r \leq n, 1 \leq p \leq n, 0 \leq b \leq 10^7$).

Các số trên cùng một dòng cách nhau bởi dấu cách.

Kết quả: Với mỗi thao tác loại 3, ghi ra thiết bị ra chuẩn một số nguyên trên một dòng là con số tại nửa trên của quân numino ở vị trí p .

Ví dụ:

Dữ liệu	Kết quả
5 30 8	17
17 18 20 22 26	10
? 1	12
< 2 4 21	4
? 3	8
? 2	
> 3 5 25	
? 5	
< 1 4 24	
? 4	

Giải thích ví dụ: Bảng dưới đây mô tả các con số trên mỗi nửa của các quân numino trong ví dụ ở trên:

Vị trí	1	2	3	4	5
Ban đầu	<div>17 13</div>	<div>18 12</div>	<div>20 10</div>	<div>22 8</div>	<div>26 4</div>
Sau thao tác 2: < 2 4 21	<div>17 13</div>	<div>12 18</div>	<div>10 20</div>	<div>22 8</div>	<div>26 4</div>
Sau thao tác 5: > 3 5 25	<div>17 13</div>	<div>12 18</div>	<div>10 20</div>	<div>22 8</div>	<div>4 26</div>
Sau thao tác 7: < 1 4 24	<div>13 17</div>	<div>18 12</div>	<div>20 10</div>	<div>8 22</div>	<div>4 26</div>

Subtask 1 (7 điểm): $n, q \leq 9000$;

Subtask 2 (16 điểm): $n, q \leq 160000$ và trong mọi thao tác loại 1 và 2, $b \leq 4$;

Subtask 3 (17 điểm): Trong mọi thao tác loại 1 và 2, $\ell = 1$ và $r = n$;

Subtask 4 (12 điểm): $n, q \leq 160000$ và mọi thao tác loại 1 và 2 đều đứng trước loại 3;

Subtask 5 (15 điểm): $a_1 \leq a_2 \leq \dots \leq a_n$;

Subtask 6 (11 điểm): $n, q \leq 160000$;

Subtask 7 (22 điểm): Không có ràng buộc gì thêm.

Bài 5. (100 điểm) Nén dữ liệu

Tham gia cuộc thi “Thiết kế thuật toán nén dữ liệu”, các thí sinh được Ban tổ chức yêu cầu xây dựng hàm nén dữ liệu và trả lời truy vấn như sau:

- Hàm nén dữ liệu: Hàm `string zip(string x)` nhận dữ liệu được truyền vào trong x là một chuỗi nhị phân độ dài n , hàm sẽ nén dữ liệu thành chuỗi y cũng là một chuỗi nhị phân, chuỗi y là chuỗi mà hàm `zip` trả về;
- Hàm trả lời truy vấn: Hàm `vector<int> answer(int n, string y)` sẽ nhận dữ liệu n là độ dài chuỗi x và chuỗi y là chuỗi nén của chuỗi x do hàm `zip` trả về. Hàm `answer` trả về một vector r có độ dài n là kết quả của n truy vấn. Giá trị $r[i]$ là kết quả cho truy vấn thứ i ($0 \leq i < n$) là truy vấn tính số lượng số 1 nhiều nhất có thể lấy được khi chọn một chuỗi con gồm $(i + 1)$ ký tự liên tiếp bất kỳ trên chuỗi x . Trả lời của hàm `answer` được chấp nhận nếu mọi $r[i]$ chênh lệch với kết quả đúng không quá 5%. Cụ thể, gọi $s[i]$ là số lượng số 1 nhiều nhất có thể lấy được khi chọn một chuỗi con gồm $(i + 1)$ ký tự liên tiếp bất kỳ trên chuỗi x thì $\frac{|s[i] - r[i]|}{s[i]} \leq 5\%$.

Yêu cầu: Lê là một thí sinh tham gia cuộc thi trên. Hãy giúp Lê thực hiện được yêu cầu của Ban tổ chức sao cho độ dài chuỗi y càng nhỏ càng tốt.

Tương tác:

- Thí sinh cần cài đặt hàm `string zip(string x)` và hàm `vector<int> answer(int n, string y)` trong file `apzip.cpp`. Trong file `apzip.cpp` thí sinh cần phải khai báo thư viện bằng dòng lệnh `#include "apziplib.h"` ở đầu file. Ngoài ra, thí sinh được phép khai báo thêm thư viện, xây dựng các hàm, sử dụng biến toàn cục khác nếu cần. File `apzip.cpp` sẽ được biên dịch cùng với thư viện `apziplib.h` (xem thêm các file mẫu trong mục đính kèm trên hệ thống để hiểu hơn về cách tương tác với hệ thống, các file này chỉ để thí sinh hiểu cách thức tương tác, không phải dùng để chấm bài).
- Lưu ý rằng, hàm `zip` và `answer` sẽ được gọi ở các tiến trình độc lập. Vì vậy, nếu hàm `zip` có sử dụng và lưu trữ dữ liệu vào các biến toàn cục, các dữ liệu này sẽ không tồn tại khi hàm `answer` được thực thi.

Chấm điểm:

- Có tất cả 5 subtask. Với mỗi subtask, gọi Q là yêu cầu mức độ nén, với mỗi test trong subtask, cách tính % điểm như sau:

- ✓ Thí sinh sẽ bị 0% điểm nếu xảy ra một trong các trường hợp: chạy sinh lỗi; tương tác sai quy cách; chạy quá thời gian; trả lời của hàm `answer` không được chấp nhận.
- ✓ Ngược lại, gọi TS là độ dài xâu y mà thí sinh nén từ xâu x , khi đó % điểm của test là:

$$\begin{cases} 100\% & \text{điểm nếu } TS \leq Q, \\ 100\% \times \frac{3 \times Q - TS}{2 \times Q} & \text{điểm nếu } Q < TS \leq 3 \times Q, \\ 0\% & \text{điểm nếu } TS > 3 \times Q. \end{cases}$$
- Với mỗi subtask, gọi e là % điểm nhỏ nhất mà thí sinh đạt được trong các test thuộc subtask, $score$ là điểm của subtask, khi đó, điểm cho subtask được tính bằng $e \times score$.

Subtask 1 (23 điểm): $n = 10^4$ và $Q = \left\lfloor \frac{81 \times n}{100} \right\rfloor$;

Subtask 2 (19 điểm): $n = 10^4$ và $Q = \left\lfloor \frac{27 \times n}{100} \right\rfloor$;

Subtask 3 (13 điểm): $n = 10^5$ và $Q = \left\lfloor \frac{9 \times n}{100} \right\rfloor$;

Subtask 4 (24 điểm): $n = 10^5$ và $Q = \left\lfloor \frac{3 \times n}{100} \right\rfloor$;

Subtask 5 (21 điểm): $n = 10^5$ và $Q = \left\lfloor \frac{n}{100} \right\rfloor$.

Bài 6. (100 điểm) Phần thưởng

Lê là người thắng cuộc trong cuộc thi “Thiết kế thuật toán nén dữ liệu” và được nhận các phần thưởng của Ban tổ chức. Ban tổ chức đã chuẩn bị n món quà, các món quà được đánh số từ 0 đến $n - 1$, món quà thứ i ($0 \leq i < n$) có giá trị là một số nguyên dương v_i và Lê được chọn lấy một số món quà nhưng tổng giá trị các món quà không được vượt quá S . Để việc chọn các món quà thêm phần thú vị, Ban tổ chức đưa ra cách thức chọn quà như sau:

- Ban tổ chức công khai giá trị của từng món quà nhưng không cho Lê biết giá trị S .
- Lê được thực hiện không quá t lượt chọn quà, ở lượt thứ r ($0 \leq r < t$), Lê cần đưa ra một dãy số p_0, p_1, \dots, p_{n-1} là hoán vị của $0, 1, \dots, n - 1$ thể hiện thứ tự ưu tiên chọn các món quà. Khi đó, Ban tổ chức sẽ cho Lê biết số món quà mà Lê có thể nhận được nếu ưu tiên lấy các món quà theo thứ tự đó. Cụ thể, Ban tổ chức sẽ cho Lê biết giá trị c lớn nhất mà $v_{p_0} + v_{p_1} + \dots + v_{p_{c-1}} \leq S$ hoặc thông báo $c = 0$ nếu không lấy được món quà nào.
- Sau khi thực hiện một số lượt không vượt quá t , Lê có thể dừng và được phép chọn các món quà ở một lượt nào đó mà tổng giá trị các món quà ở lượt đó là lớn nhất trong các lượt đã thực hiện.

Yêu cầu: Hãy giúp Lê đưa ra cách chọn quà sao cho tổng giá trị các món quà được nhận là lớn nhất.

Tương tác:

- Thí sinh cần cài đặt hàm: `void run(int n, vector<int> v, int t)` đặt trong file `bonus.cpp`, trong đó: n là số món quà; v là vector chứa thông tin về giá trị các món quà với $v[i]$ là giá trị món quà thứ i ($0 \leq i < n$); t là số lượt cho phép. Trong file `bonus.cpp` thí sinh cần phải khai báo thư viện bằng dòng lệnh `#include "bonuslib.h"` ở đầu file. Ngoài ra, thí sinh được phép khai báo thêm thư viện, xây dựng các hàm, sử dụng biến toàn cục khác nếu cần. File `bonus.cpp` sẽ được biên dịch cùng với thư viện `bonuslib.h` (xem thêm các file mẫu trong mục đính kèm

trên hệ thống để hiểu hơn về cách tương tác với hệ thống, các file này chỉ để thí sinh hiểu cách thức tương tác, không phải dùng để chấm bài).

- Hàm `run` không được gọi quá t lần hàm `select` (hàm đã được xây dựng trong `bonuslib.h`). Hàm `int select(vector<int> p)`, trong đó p là một hoán vị của $0, 1, \dots, n-1$, sẽ trả về số lượng món quà lấy được nếu ưu tiên lấy các món quà theo thứ tự mô tả bằng hoán vị p .

Chấm điểm:

- Có tất cả 6 subtask. Với mỗi test trong mỗi subtask, luôn tồn tại phương án chọn để tổng giá trị bằng đúng S và % điểm cho mỗi test được tính như sau:
 - Thí sinh sẽ bị 0% điểm nếu xảy ra một trong các trường hợp: chạy sinh lỗi; tương tác sai quy cách; chạy quá thời gian; số lần gọi hàm `select` vượt quá t hoặc vector p truyền vào hàm `select` không phải là một hoán vị của $0, 1, \dots, n-1$;
 - Ngược lại, gọi TS là tổng giá trị của các món quà trong cách chọn của thí sinh, GK là tổng giá trị của các món quà trong cách chọn của giám khảo, đặt $Q = 2 \times n \times \frac{GK - TS}{GK}$, khi đó % điểm của test là:

$$\begin{cases} 0\% & \text{điểm nếu } Q > 1, \\ 100\% & \text{điểm nếu } Q < 0, \\ 100\% \times (-\log_{10}(0.9 \times Q + 0.1)) & \text{điểm nếu } 0 \leq Q \leq 1. \end{cases}$$

- Với mỗi subtask, gọi e là % điểm nhỏ nhất mà thí sinh đạt được trong các test thuộc subtask, $score$ là điểm của subtask, khi đó, điểm cho subtask được tính bằng $e \times score$.

Subtask 1 (10 điểm): $n \leq 6; t = 32$ và $v_i \leq 10^3$;

Subtask 2 (10 điểm): $n \leq 100; t = 32$ và $v_i \leq 10^3$;

Subtask 3 (20 điểm): $n \leq 500; t = 5$ và $v_i \leq 10^3$;

Subtask 4 (10 điểm): $n \leq 100; t = 32$ và $v_i \leq 10^9$;

Subtask 5 (20 điểm): $n \leq 500; t = 32$ và $v_i \leq 10^9$;

Subtask 6 (30 điểm): $n \leq 500; t = 5$ và $v_i \leq 10^9$.

Ví dụ:

Giả sử có 4 món đồ với giá trị tương ứng là $\{9, 3, 1, 5\}$ và $t = 3, S = 8$, hàm `run` sẽ được gọi với giá trị các tham số tương ứng là `run(4, {9, 3, 1, 5}, 3)`, dưới đây là một ví dụ cách gọi lần lượt các hàm `select`:

Lời gọi hàm	Kết quả trả về	Giải thích
<code>select({0, 1, 2, 3})</code>	0	Không lấy được món quà nào
<code>select({1, 3, 0, 2})</code>	2	Lấy được 2 món quà số 1 và số 3 với tổng giá trị là 8
<code>select({1, 2, 3, 0})</code>	2	Lấy được 2 món quà số 1 và số 2 với tổng giá trị là 4

Với cách gọi trên, thí sinh sẽ chọn được 2 món quà với tổng giá trị là 8 và đạt 100% số điểm của test đó.

----- HẾT -----

- Thí sinh không được sử dụng tài liệu.
- Giám thị không giải thích gì thêm.