

SỞ GIÁO DỤC VÀ ĐÀO TẠO
THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG THPT CHUYÊN
LÊ HỒNG PHONG



ĐỀ CHÍNH THỨC

KỲ THI OLYMPIC TRUYỀN THÔNG 30 THÁNG 4
LẦN THỨ XXVII - NĂM 2023
Ngày thi: 08/4/2023
MÔN THI: TIN HỌC - KHỐI: 10
THỜI GIAN: 180 phút
Hình thức làm bài: Tự luận
Đề thi có 03 trang

Tổng quan về đề thi

Câu	Tên bài	File dữ liệu vào	File kết quả	Tên chương trình
1	Số cách đi quân mã	SOCACH.INP	SOCACH.OUT	SOCACH.*
2	Software	SOFTWARE.INP	SOFTWARE.OUT	SOFTWARE.*
3	Chứng khoán	CHUNGKHOAN.INP	CHUNGKHOAN.OUT	CHUNGKHOAN.*

Dấu * thay thế bởi PAS hoặc CPP của ngôn ngữ lập trình được sử dụng tương ứng là Pascal hay C++

Câu 1. (6 điểm) Số cách đi quân mã

Xét một bàn cờ vua có kích thước $m \times n$ gồm có m dòng, n cột.

Các dòng được đánh số từ 1 đến m , các cột được đánh số từ 1 đến n .

Một ô nằm trên dòng x , cột y được kí hiệu là (x, y) .

Một quân mã xuất phát từ một ô trên bàn cờ có thể đi đến một trong bốn ô như hình vẽ.

Ngoài ra, trên bàn cờ có k ô mà quân mã không được phép đi vào. Những ô này được gọi là ô bị cấm.

Yêu cầu: Tìm số cách di chuyển của quân mã từ ô (x, y) cho trước đến ô (m, n) .

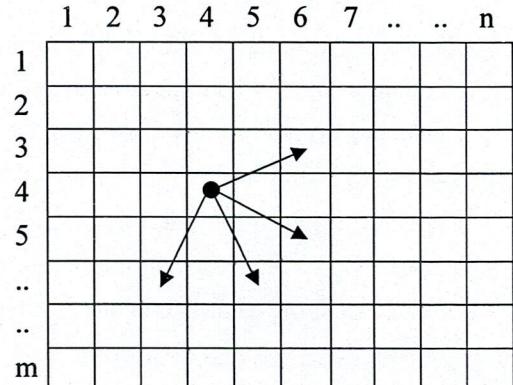
Dữ liệu vào: Cho trong file văn bản **SOCACH.INP** gồm:

- Dòng đầu tiên ghi 4 số nguyên m, n, k và Q ($1 < m, n \leq 1000 ; 0 \leq k \leq 10 ; Q \leq 10^5$)
- Trên k dòng tiếp theo, mỗi dòng ghi 2 số nguyên kx_i, ky_i cho biết ô (kx_i, ky_i) bị cấm ($1 \leq kx_i < m ; 1 \leq ky_i < n$)
- Trên Q dòng tiếp theo, mỗi dòng ghi 2 số nguyên x_i, y_i ($1 \leq x_i < m ; 1 \leq y_i < n$).

Các số trên một dòng cách nhau bởi ít nhất một khoảng trắng.

Kết quả: Đưa ra file văn bản **SOCACH.OUT**, gồm Q số nguyên W_i , mỗi số trên một dòng cho biết số cách di chuyển quân mã từ ô (x_i, y_i) cho trước đến ô (m, n) .

Trong đó W_i là phần dư của phép chia số cách quân mã di chuyển từ ô (x_i, y_i) cho trước đến ô (m, n) cho 10^9 .



Ví dụ:

SOCACH.INP	SOCACH.OUT	SOCACH.INP	SOCACH.OUT
4 5 1 3	1	4 5 0 2	2
2 4	1	1 2	1
1 2	0	3 3	
3 3			
2 4			

Ràng buộc:

- 30% test ứng với 30% số điểm của bài có $1 < m, n \leq 100 ; k = 0; Q \leq 10^2$
- 20% test ứng với 20% số điểm của bài có $1 < m, n \leq 100 ; 0 < k \leq 10; Q \leq 10^2$
- 50% test ứng với 50% số điểm của bài có ràng buộc như đề bài.

Câu 2. (7 điểm) Software

Tâm rất yêu thích lập trình tạo phần mềm. Vào dịp rảnh rỗi Tâm đã thiết kế một phần mềm đơn giản. Màn hình phần mềm gồm N địa điểm (đánh số từ 1 đến N), trong đó mỗi địa điểm có đặt một bóng đèn ở trạng thái sáng hoặc tắt. Có N-1 con đường một chiều nối trực tiếp giữa các cặp địa điểm. Mỗi lần Tâm chạm tay vào một địa điểm X_i bất kì trên màn hình thì sẽ có một robot xuất phát từ địa điểm X_i di chuyển theo các con đường một chiều, cuối cùng kết thúc ở địa điểm 1. Robot *không* thay đổi trạng thái đèn ở địa điểm X_i và địa điểm 1, các địa điểm còn lại robot đã đi qua thì đèn ở địa điểm đó sẽ thay đổi sang trạng thái ngược lại (sáng thành tắt, tắt thành sáng).

Yêu cầu: Hãy cho biết khi Tâm thực hiện K lần chạm tay (mỗi lần chạm tay vào một địa điểm) thì sau đó sẽ có tất cả bao nhiêu địa điểm có đèn sáng. Biết rằng robot xuất phát từ địa điểm bất kì luôn có thể di chuyển theo các con đường một chiều đến địa điểm 1.

Dữ liệu: Cho trong file SOFTWARE.INP

- Dòng đầu tiên gồm 2 số nguyên dương N và K lần lượt là số địa điểm, số lần chạm tay ($1 \leq N, K \leq 100000$).
- Dòng thứ hai gồm N số nguyên cho biết trạng thái đèn ở N địa điểm, lần lượt theo thứ tự từ địa điểm 1 đến địa điểm N. Trạng thái đèn tắt là 0, sáng là 1.
- Dòng thứ i trong N-1 dòng tiếp theo gồm hai số nguyên dương A_i và B_i ($1 \leq A_i, B_i \leq N$) cho biết có con đường một chiều nối trực tiếp từ địa điểm A_i đến B_i .
- Dòng cuối cùng gồm K số nguyên dương, trong đó số nguyên thứ i là X_i ($1 \leq X_i \leq N$) cho biết địa điểm thứ i mà Tâm thực hiện chạm tay.

Kết quả: Ghi ra file SOFTWARE.OUT số nguyên duy nhất là kết quả cần tìm.

Ví dụ:

SOFTWARE.INP	SOFTWARE.OUT
5 3	
1 0 0 0 0	3
2 1	
4 2	
3 2	
5 4	
4 5 4	

SOFTWARE.INP	SOFTWARE.OUT
4 1	
0 0 0 0	2
3 1	
2 3	
4 2	
4	

Giải thích ví dụ 1: Tâm chạm tay 3 lần.

Lần 1 ở địa điểm 4: robot đi qua các địa điểm 4, 2, 1 và thay đổi trạng thái đèn ở địa điểm 2.

Kết quả trạng thái 5 đèn theo thứ tự lần lượt là: 1 1 0 0 0

Lần 2 ở địa điểm 5: robot đi qua các địa điểm 5, 4, 2, 1 và thay đổi trạng thái đèn ở địa điểm 4, 2.

Kết quả trạng thái 5 đèn lần lượt là: 1 0 0 1 0

Lần 3 ở địa điểm 4: robot đi qua các địa điểm 4, 2, 1 và thay đổi trạng thái đèn ở địa điểm 2.

Kết quả trạng thái 5 đèn lần lượt là: 1 1 0 1 0

Sau 3 lần chạm tay, có 3 địa điểm có đèn sáng là địa điểm 1, 2 và 4.

Ràng buộc: 50% số test ứng với 50% số điểm có $1 \leq N, K \leq 5000$.

Câu 3. (7 điểm) Chứng khoán

Chứng khoán là một trong những kênh đầu tư nhiều rủi ro. Nhà đầu tư cần phân tích nhiều yếu tố thị trường để quyết định đầu tư vào một cổ phiếu. Một trong những tham số được nhiều nhà đầu tư quan tâm là tính ổn định giá của một cổ phiếu. Xét giá bán của một cổ phiếu trong nhiều phiên giao dịch. Cổ phiếu được xem là có giá ổn định trong một khoảng thời gian nếu khoảng chênh lệch của giá bán cao nhất và giá bán thấp nhất của cổ phiếu đó trong khoảng thời gian trên không vượt quá ngưỡng ổn định giá T cho trước.

Yêu cầu: Cho biết giá bán một cổ phiếu trong N phiên giao dịch. Hãy viết chương trình tính số phiên giao dịch dài nhất mà cổ phiếu đó có giá ổn định.

Dữ liệu: Vào từ file văn bản CHUNGKHOAN.INP dòng đầu chứa hai số nguyên $T (0 \leq T \leq 2.10^9)$ và N lần lượt cho biết ngưỡng ổn định giá và số phiên giao dịch. Dòng thứ hai chứa N số nguyên $G_i (0 \leq G_i \leq 2.10^9)$ lần lượt cho biết giá bán của cổ phiếu trong N phiên giao dịch.

Kết quả: Ghi ra file văn bản CHUNGKHOAN.OUT một số nguyên là số phiên giao dịch dài nhất mà cổ phiếu có giá ổn định.

Ví dụ:

CHUNGKHOAN.INP	CHUNGKHOAN.OUT
5 10	
5 7 9 20 15 13 20 12 11 1	3

Ràng buộc:

- 30% test ứng với 20% số điểm của bài có $1 < N \leq 10\,000$
- 20% test ứng với 30% số điểm của bài có $10\,000 < N \leq 500\,000$
- 50% test ứng với 50% số điểm của bài có $500\,000 < N \leq 3\,000\,000$

----- HẾT -----
Cán bộ coi thi không giải thích gì thêm.

Họ tên thí sinh: SBD:

Trường: Tỉnh/TP:



Bài	Nội dung	Điểm
Bài 1	<p>SỐ CÁCH ĐI QUÂN MÃ</p> <p>-Có 10 test, mỗi test 0.6 điểm</p> <p>-Phân bố test như ràng buộc ở đề bài</p> <p>-Đáp án:</p> <p>Lời giải có $O(Q.(m+n).n)$:</p> <p>-Úng với mỗi truy vấn (x_i, y_i) ta tìm số cách quân mã di chuyển từ ô (x_i, y_i) cho trước đến ô (m, n).</p> <p>Việc tìm số cách quân mã di chuyển từ ô (x_i, y_i) cho trước đến ô (m, n) có thể sử dụng phương pháp duyệt BFS từ ô (x_i, y_i) hoặc Quy hoạch động. Thao tác này mất $O((m+n).n)$</p> <p>Tổng cộng sẽ có $O(Q.(m+n).n)$</p> <p>Lời giải có $O(Q+(m+n).n)$:</p> <p>Nhận xét:</p> <p>Nếu có cách đi từ ô (x_i, y_i) đến ô (m,n) theo một tuyến đường nào đó thì sẽ có cách đi cách đi từ ô (m,n) đến ô (x_i, y_i) theo chiều ngược lại (cũng theo tuyến đường này).</p> <p>Do đó ta thực hiện như sau:</p> <p>-Dùng BFS hoặc quy hoạch động để tìm số cách $SC[i,j]$ đi từ ô (m,n) đến ô (x_i, y_i) bất kỳ của bàn cờ . Thao tác này mất $O((m+n).n)$.</p> <p>-Úng với mỗi truy vấn (x_i, y_i) ta xuất $S[i,j]$ nên mất $O(1)$.</p> <p>Với Q truy vấn sẽ mất $O(Q)$.</p> <p>Tổng cộng sẽ có $O(Q+(m+n).n)$</p> <p>Lưu ý: Trong quá trình tính $SC[i,j]$ ta có sử dụng phép toán lấy phần dư của phép chia cho 10^9.</p>	6
Bài 2	<p>SOFTWARE</p> <p>-Có 10 test, mỗi test 0.7 điểm</p> <p>-Phân bố test như ràng buộc ở đề bài</p> <p>-Đáp án:</p> <p>Bài toán có dạng đồ thị dưới dạng cây (N đỉnh, $N-1$ cạnh) có gốc tại đỉnh 1.</p> <p>Trường hợp $1 \leq N, K \leq 5000$</p> <ul style="list-style-type: none"> Gọi $parent[v]$ là đỉnh thăm lién kè trước v. Thực hiện DFS(1) để xác định $parent[v]$ của tất cả các đỉnh v trong cây. Với mỗi đỉnh x Tâm chạm tay, robot sẽ xuất phát từ x đi đến các đỉnh thăm trước x ($parent[x]$) đến khi gặp đỉnh 1. Với mỗi đỉnh robot đi qua (ngoại trừ đỉnh 1 và đỉnh xuất phát x), thực hiện thay đổi trạng thái đèn 	7

Bài	Nội dung	Điểm
	<p>của đỉnh.</p> <ul style="list-style-type: none"> Sau khi thực hiện K lần chạm tay, đếm số đèn sáng. <p>Trường hợp đạt điểm tối đa, thực hiện quy hoạch động. Gọi $dp[u]$ là số lần thay đổi trạng thái của đèn tại đỉnh u</p> <ul style="list-style-type: none"> $dp[u] =$tổng tất cả các $dp[v]$, với v là các đỉnh kề với u. Thực hiện duyệt DFS(1) để tính $dp[u]$ <p>Trạng thái đèn tại một địa điểm u (với u khác 1) sáng trong hai trường hợp sau:</p> <ul style="list-style-type: none"> trạng thái đèn tại u ban đầu là 1 và $dp[u]$ là số chẵn trạng thái đèn tại u ban đầu là 0 và $dp[u]$ là số lẻ 	
Bài 3	<p>CHÚNG KHOÁN</p> <p>-Có 10 test, mỗi test 0.7 điểm</p> <p>-Phân bố test như ràng buộc ở đề bài</p> <p>-Đáp án:</p> <p>Thuật Toán $O(N^3)$</p> <p>Xét từng đoạn phiên giao dịch có thể có, với mỗi đoạn tìm giá trị lớn nhất, nhỏ nhất của từng đoạn, kiểm tra tính ổn định của đoạn, cập nhật đoạn ổn định dài nhất.</p> <p>Thuật toán $O(N^2)$</p> <p>Nhận xét nếu đoạn (i,j) đã không ổn định thì đoạn (i',j') với $i' \leq i$ hay $j \leq j'$ cũng không ổn định. Ta có thể cố định một đầu của đoạn trong quá trình tìm đoạn ổn định dài nhất.</p> <p>Thuật toán $O(N \log N)$</p> <p>Dùng kỹ thuật hai con trỏ để biểu diễn hai đầu của đoạn ổn định. Nếu $a[j] (j > i)$ làm cho đoạn đang xét không ổn định ta tăng i lên, ngược lại tăng j. Để tối ưu quá trình tìm số lớn nhất hay nhỏ nhất trong một đoạn, ta có thể dùng các cấu trúc quản lý đoạn (segment tree) hay dùng multiset để có thể tìm giá trị lớn nhất, nhỏ nhất với độ phức tạp $O(\log N)$.</p> <p>Thuật toán $O(N)$</p> <p>Gọi $p[j]$ cho biết chỉ số i nhỏ nhất để đoạn $[i, j]$ ổn định, $\min[i, j]$: giá trị nhỏ nhất trong đoạn $[i, j]$, $\max[i, j]$: giá trị lớn nhất trong đoạn $[i, j]$.</p> <p>Giả sử ta đã biết $p[j-1]$</p> <p>$p[j]$ chỉ khác $p[j-1]$ khi $(a[j] - \min[p[j-1], j-1]) > T$ hoặc $(\max[p[j-1], j-1] - a[j]) > T$.</p> <p>Nếu $a[j-1] > a[j]$. Ta cần lưu chỉ số <i>dãy số giảm dần dài nhất</i> có phần tử cuối là $a[j]$.</p> <p>Nếu $a[j-1] < a[j]$. Ta cần lưu chỉ số <i>dãy số tăng dần dài nhất</i> có phần tử cuối là $a[j]$.</p>	7

Bài	Nội dung									Điểm	
	<p>Nếu $a[j-1] = a[j]$. Ta cần chỉ cập nhật $p[j]=p[j-1]$</p> <p>Xét ví dụ với dãy sau có $T=4$ và đoạn ổn định dài nhất hiện tại là 4</p>										
	1	2	3	4	5	6	7	8	9		
	5	4	6	<u>8</u>	7	<u>7</u>	4	<u>6</u>	4		
	<p>Ta có:</p> <ul style="list-style-type: none"> • dãy số giảm dần dài nhất là 8, 7, 6, 5 với các chỉ số tương ứng là 4, 6, 8, 10 • dãy số tăng dần dài nhất là 4, 5 với các chỉ số tương ứng là 9, 10 • $p[10]=1$ <p>Nếu thêm $a[11] = 2$, Ta có</p> <ul style="list-style-type: none"> • $p[11]=7$ • dãy số giảm dần dài nhất cập nhật lại là 6, 5, 2 với các chỉ số tương ứng là 8, 10, 11 • dãy số tăng dần dài nhất cập nhật lại là 2 với các chỉ số tương ứng là 11 <p>Với hai dãy tăng/giảm dần này, ta có thể cập nhật lại $p[j]$ một cách dễ dàng. Nhận thấy rằng những giá trị trong dãy giảm mà lớn hơn $a[j]$ T đơn vị sẽ bị loại bỏ trong dãy mới và $p[j]$ sẽ được cập nhật là $p1=[(chi số của phần tử cuối trong dãy giảm lớn hơn a[j] T đơn vị) +1]$. Xét tương tự cho dãy tăng để có $p2$ và $p[j]$ mới = $\max(p1, p2)$</p>										