

Bài 1: Dây chuyền sản xuất

Một nhà máy chạy một dây chuyền sản xuất. Có 2 nguyên công (2 giai đoạn độc lập nối tiếp nhau) cần phải thực hiện đối với mỗi một sản phẩm theo trình tự sau: Đầu tiên thực hiện nguyên công A, sau đó thực hiện nguyên công B. Có một số máy để thực hiện từng nguyên công (như vậy có hai loại máy: Máy thực hiện nguyên công A - *máy kiểu A* và máy thực hiện nguyên công B - *Máy kiểu B*). Dây chuyền sản xuất thực hiện như sau:

Máy kiểu A lấy sản phẩm từ băng chuyền vào, thực hiện nguyên công A và đặt sản phẩm vào băng chuyền trung gian. Máy kiểu B lấy sản phẩm từ băng chuyền trung gian, thực hiện nguyên công B và đặt sản phẩm vào băng chuyền ra. Mọi máy đều có thể làm việc song song và độc lập nhau, mỗi máy làm việc với thời gian xử lý cho trước. Thời gian xử lý là số đơn vị thời gian cần thiết để thực hiện nguyên công bao gồm cả thời gian lấy sản phẩm từ băng chuyền trước khi xử lý và thời gian đặt sản phẩm lên băng chuyền sau khi xử lý

Câu a: Đưa ra thời điểm sớm nhất nguyên công A được hoàn thành đối với tất cả N sản phẩm với điều kiện là các sản phẩm này đã sẵn sàng trên băng chuyền vào tại thời điểm 0

Câu b: Đưa ra một thời điểm sớm nhất mà cả hai nguyên công A và B được hoàn thành đối với tất cả N sản phẩm khi các sản phẩm này đã sẵn sàng trên băng chuyền vào thời điểm 0

Dữ liệu: Vào từ file văn bản ASSLINE.INP gồm các số nguyên dương ghi trên 5 dòng

- Dòng thứ nhất ghi N số là số sản phẩm ($1 \leq N \leq 1000$)
- Dòng thứ hai ghi M_1 là số lượng các máy kiểu A ($1 \leq M_1 \leq 30$)
- Dòng thứ ba ghi M_1 số nguyên là các thời gian xử lý của từng máy kiểu A
- Dòng thứ tư và thứ năm tương ứng ghi M_2 là số lượng các máy kiểu B ($1 \leq M_2 \leq 30$) và thời gian xử lý của từng máy kiểu B. Thời gian xử lý là một số nguyên nằm trong khoảng từ 1 đến 20

Kết quả: Ghi ra file văn bản ASSLINE.OUT trên 2 dòng

- Dòng đầu tiên chứa một số nguyên dương là lời giải của câu a
- Dòng thứ hai là một số nguyên dương là lời giải của câu b

Ví dụ:

ASSLINE . INP	ASSLINE . OUT
5	3
2	5
1 1	
3	
3 1 4	

Bài 2: Bưu cục

Có một làng nằm dọc theo một đường cao tốc. Đường cao tốc được biểu diễn bằng một trục số nguyên và vị trí mỗi làng được xác định bởi một số nguyên duy nhất. Không có hai làng

nào ở cùng một vị trí. Khoảng cách giữa hai vị trí bằng giá trị tuyệt đối của hiệu nằm giữa hai toạ độ nguyên của chúng.

Một số bưu cục được xây dựng ở một số làng nhưng không nhất thiết tại mọi làng. Mỗi làng và bưu cục thuộc nó có cùng vị trí. Để xây dựng các bưu cục, vị trí của chúng cần chọn sao cho tổng các khoảng cách từ mỗi làng đến bưu cục gần nhất đối với làng đó là nhỏ nhất

Bạn cần viết chương trình sao cho khi biết các vị trí của các làng và số lượng các bưu cục, hãy tìm tổng nhỏ nhất có thể được của các khoảng cách từ mỗi làng đến bưu cục gần nhất đối với nó và các vị trí tương ứng của các bưu cục

Dữ liệu vào: Trong file POST.INP

- Dòng thứ nhất chứa hai số nguyên: số thứ nhất là số làng v , $1 \leq v \leq 300$, số thứ hai là số lượng bưu cục p , $1 \leq p \leq 30$, $p \leq v$
- Dòng thứ hai chứa v số nguyên theo thứ tự tăng dần, v số nguyên này là các vị trí của v làng. Với mỗi vị trí x , $1 \leq x \leq 10000$

Dữ liệu ra trong file POST.OUT.

- Dòng thứ nhất chứa một số nguyên s là tổng nhỏ nhất có thể được của các khoảng cách từ mỗi làng đến bưu cục gần nhất đối với nó theo thông báo trong dòng thứ hai.
- Dòng thứ hai chứa P số nguyên theo thứ tự tăng dần. Các số nguyên này là các vị trí của các làng khác nhau tại đó đặt bưu cục. Có thể có một số lời giải, chương trình của bạn chỉ cần đưa ra một

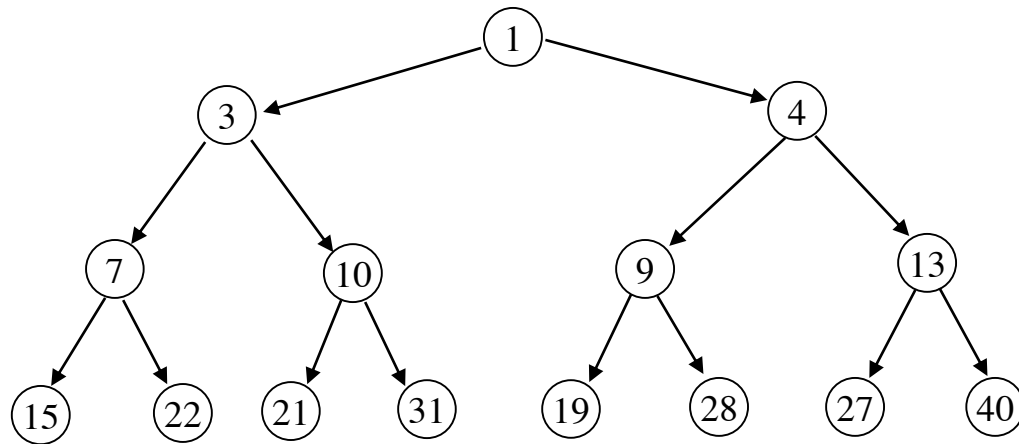
Ví dụ:

POST . INP	POST . OUT
10 5	9
1 2 3 6 7 9 11 22 44 50	2 7 22 44 50

Bài 3: Tập N

Xét tập số nguyên s , ban đầu tập chỉ chứa số 1. Người ta biến đổi S theo cách sau:

Thay thế mỗi số nguyên x trong S bằng 2 số mới $2x+1$ và $3x+1$



Yêu cầu: Cho 2 số nguyên N và M ($1 \leq N \leq 31$, $1 \leq M \leq 2^N$). Sau N lần biến đổi S , hãy xác định số thứ M của S , nếu các số trong S được sắp xếp theo thứ tự tăng dần.

Dữ liệu: Vào từ file văn bản TAP_N.INP gồm nhiều dòng, mỗi dòng chứa 2 số nguyên N và M (cách nhau ít nhất một dấu cách)

Kết quả: Đưa ra file văn bản TAP_N.OUT, mỗi kết quả là một số nguyên và đưa ra trên một dòng

Ví dụ:

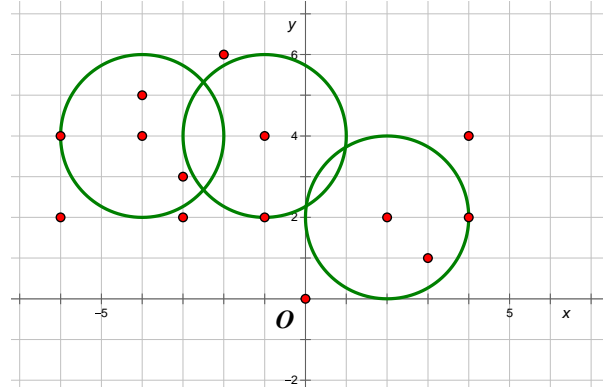
TAP N . INP	TAP N . OUT
2 4	13
3 2	19

Câu 4. (7,0 điểm) Bắt ruồi

Ở xứ ManU ngày nay có một người nổi tiếng tên là Anthony Martial. Anh được mệnh danh là Tia chớp đen nhờ có khả năng bắt ruồi rất nhanh và chính xác. Martial thường quan sát vị trí các chú ruồi rồi tung một cú đập trúng một lúc vài ba con liền.

Trên mặt bàn được xem là mặt phẳng Oxy, có N con ruồi, con thứ i ở vị trí tọa độ (x_i, y_i) và không có 2 con ruồi nào ở cùng một chỗ.

Martial dùng một chiếc vĩ đập ruồi hình tròn có bán kính R và đập liên tiếp M lần xuống bàn. Ruồi bị bắt nếu vĩ đập ruồi phủ lên vị trí nó đang ở (xem hình vẽ).



Yêu cầu: Hãy đếm số ruồi mà Martial bắt được sau M lần đập. Biết rằng khoảng thời gian giữa các lần đập ruồi là không đáng kể, các chú ruồi không kịp bay đi sau mỗi lần đập.

Dữ liệu: Vào từ tệp văn bản “SPID.INP”:

- Dòng đầu tiên chứa số nguyên dương N, M, R.
- Dòng thứ i trong N dòng tiếp theo chứa 2 số nguyên x_i, y_i các tọa độ là nơi những con ruồi đang đậu.
- Dòng thứ j trong M dòng tiếp theo chứa 2 số nguyên z_j, t_j các tọa độ là tâm của vĩ đập ruồi lần thứ j.

SPID.INP
10 3 2
-6 2
-6 4
-4 5
-3 2
-3 3
-2 6
-1 2
3 1
4 2
4 4
-4 4
-1 4
2 2

SPID.OUT
6

Với $-10^5 \leq x_i, y_i, z_j, t_j \leq 10^5$; $R \leq 10^9$.

Kết quả: Đưa ra tệp văn bản “SPID.OUT” số nguyên duy nhất là số ruồi mà Martial bắt được sau M lần đập.

Subtask#01: 60% số điểm của bài tương ứng với $N, M \leq 10^3$.

Subtask#02: 40% số điểm của bài tương ứng với $N \leq 10^3$; $M \leq 10^6$.

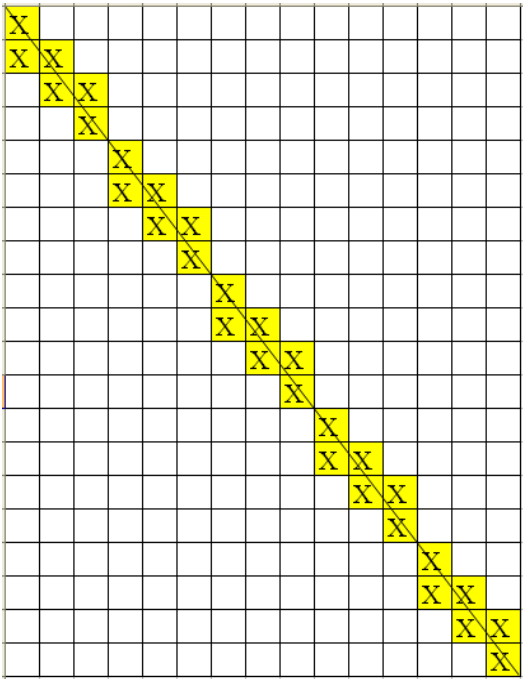
Câu 5. (7,0 điểm) Sô cô la

Jack tặng Oggy một miếng Sôcôla hình chữ nhật kích thước $M \times N$ chia thành các ô vuông 1×1 . Oggy không thưởng thức miếng Sôcôla một mình mà chia miếng Sôcôla này cho những bạn gián ở cùng nhà. Oggy kẻ một đường chéo từ góc trái trên đến góc phải dưới của miếng Sôcôla hình chữ nhật trên và giữ lại những miếng Sôcôla nhỏ, hình vuông 1×1 mà đường chéo cắt qua.

Yêu cầu: Đếm số miếng Sôcôla hình vuông 1×1 mà Oggy giữ lại.

Dữ liệu: Vào từ tệp văn bản “SOCOLA.inp”: gồm hai số nguyên dương M và N là kích thước miếng Sôcôla hình chữ nhật.

Kết quả: Ghi ra tệp văn bản “SOCOLA.out”: số nguyên duy nhất là số lượng miếng Sôcôla nhỏ hình vuông 1×1 mà Oggy giữ lại.



Miếng Sôcôla kích thước c
 20×15

Ví dụ:

SOCOLA.inp	SOCOLA.out	Hình minh họa
20 15	30	Những miếng Sôcôla Oggy giữ lại được đánh dấu [x].

Subtask#01: 40% số điểm của bài ứng với $0 < M, N \leq 10^5$.

Subtask#02: 60% số điểm của bài ứng với $0 < M, N \leq 10^9$.

Câu 6. (6,0 điểm) Trò chơi SASUKE

Vừa sang tới Việt Nam, Martial được Ban biên tập VTV3 mời tham dự trò chơi Vượt chướng ngại vật SASUKE. Ở trò chơi này, Martial phải vượt qua được N chướng ngại vật là các cột có độ cao $H[i]$. Với thân hình khỏe mạnh do được rèn luyện thường xuyên, anh ấy chỉ có thể nhảy từ cột thứ i sang cột thứ $i+1$ với chi phí năng lượng là $|H[i]-H[i+1]|$ hoặc nhảy từ cột thứ i sang cột thứ $i+2$ với chi phí năng lượng là $2 \cdot |H[i]-H[i+2]|$.

$H[i+2]$. Martial đang đứng ở cột 0 – cột xuất phát. Ở bước nhảy đầu tiên anh ấy bắt buộc phải nhảy sang cột 1.

Ví dụ: Có $N = 6$ cột, có độ cao là 6, 8, 4, 6, 4, 5. Martial có thể:

- Nhảy theo thứ tự cột (1-2-4-6), chi phí năng lượng là: $6+2+2*|8-6|+2*|6-5| = 14$.
- Hoặc nhảy theo thứ tự cột (1-3-5-6), chi phí năng lượng là: $6+2*|6-4|+2*|4-4|+|4-5| = 11$.

Yêu cầu: Hãy tính chi phí năng lượng ít nhất mà Martial phải dùng để vượt qua được N chướng ngại vật của trò chơi SASUKE.

Dữ liệu: Vào từ tệp “GSASUKE.INP” gồm:

- Dòng đầu tiên ghi số nguyên dương N .
- Dòng thứ hai ghi N số nguyên dương $H[i]$ là độ cao của cột chướng ngại vật thứ i .

Kết quả: Ghi ra tệp văn bản “GSASUKE.OUT” số nguyên duy nhất là chi phí năng lượng ít nhất mà Martial phải dùng để vượt qua N chướng ngại vật.

Subtask#01: 30% số điểm của bài tương ứng với $N \leq 20$.

Subtask#02: 70% số điểm của bài tương ứng với $N \leq 10^5$.

GSASUKE.INP
5
2 1 4 3 1
6
6 8 4 6 4 5

GSASUKE.OUT
9
11