**BÀI TẬP CHẶT NHỊ PHÂN**

**Bài 1: ASSIGN1 - Phân công hoàn thành sớm nhất**

*Đề bài:* Có n người, n việc (1 < n ≤ 200). Người thứ i thực hiện công viêc j mất C[i,j] đơn vị thời gian. Giả sử tất cả bắt đầu vào thời điểm 0, hãy tìm cách bố trí mỗi công việc cho mỗi người sao cho thời điểm hoàn thành công việc là sớm nhất có thể.

*Input:*

- Dòng đầu: N .

- Tiếp theo là ma trận C[i,j]. (thuộc kiểu Integer)

*Output:* Ghi thời điểm sớm nhất hoàn thành.

*Ví dụ:*

Input

4

10 10 10 2

10 10 3 10

4 10 10 10

10 5 10 10

Output

5

*Gợi ý giải*:

- Sử dụng phương pháp chặt nhị phân và ghép cặp cực đại trong đồ thị 2 phía.

- Yêu cầu bài toán là ta cần tìm thời gian ít nhất để n người thực hiện xong n việc, mỗi người được phân công 1 việc. Cách đơn giản nhất là ta chỉ cần duyệt từ khoảng thời gian thấp nhất đến cao nhất, tìm thời điểm đầu tiên n người làm xong n việc, mỗi người 1 việc. Với thời gian đáng xét cụ thể, ta có thể phân việc bằg thuật toán ghép cặp trên đồ thị 2 phía với tập X là người, tập Y là công viêc, cạnh C[i][j] thuộc đồ thị 2 phía nếu C[i][j] thời gian đang xét. Thời gian càng tăng thì số cặp ghép cực đại không giảm, do đó ta tiến hành chặt nhị phân trong khoảng thời gian [thấp nhất .. cao nhất] để tìm thời gian sớm nhất thỏa mãn yêu cầu bài toán.

- Độ phức tạp: n3log(Cmax).

**Bài 2: MTWALK - Mountain Walking**

*Đề bài:* Cho một bản đồ kích thước NxN (2 <= N <= 100), mỗi ô mang giá trị là độ cao của ô đó (0 <= độ cao <= 110). Bác John và bò Bessie đang ở ô trên trái (dòng 1, cột 1) và muốn đi đến cabin (dòng N, cột N). Họ có thể đi sang phải, trái, lên trên và xuống dưới nhưng không thể đi theo đường chéo. Hãy giúp bác John và bò Bessie tìm đường đi sao cho chênh lệch giữa điểm cao nhất và thấp nhất trên đường đi là nhỏ nhất.

*Input:*

- Dòng 1: N.

- Dòng 2 .. N+1: Mỗi dòng chứa N số nguyên, mỗi số cho biết cao độ của một ô.

*Output:* Một số nguyên là chênh lệch cao độ nhỏ nhất.

*Ví dụ:*

Input

5

1 1 3 6 8

1 2 2 5 5

4 4 0 3 3

8 0 2 3 4

4 3 0 2 1

Output

2

*Gợi ý giải:*

- Sử dụng phương pháp BFS (tìm kiếm theo chiều rộng) và chặt nhị phân.

- Giả sử đường đi từ (1,1) → (n,n) sẽ đi qua đỉnh có độ cao lớn nhất là hmax và đỉnh có độ cao nhỏ nhất là hmin, ta cần tìm min(hmax - hmin). Ta duyệt giá trị hmin, hmax, với mỗi cặp hmax, hmin ta kiểm ta từ (1,1) đi đến (n,n) hay không thông qua các đỉnh có độ cao từ [hmin .. hmax], trong tất cả kết quả từ (1,1) đi đến được (n,n) ta lấy ra kêqt quả hmax – hmin nhỏ nhất. Vói mỗi hmin đang xét, thay vì duyệt tiếp hmax, ta sẽ chặt nhị phân tìm hmax nhỏ nhất mà (1,1) đến được (n,n). Quá trình đi từ (1,1) → (n,n) sử dụng BFS (tìm kiếm theo chiểu rộng) để kiểm tra.

- Độ phức tạp: n2hlog(h) (h là độ cao lớn nhất).

**Bài 3: LIS - Dãy con tăng dài nhất**

*Đề bài:* Cho một dãy gồm N số nguyên (1 ≤ N ≤ 30000). Hãy tìm dãy con tăng dài nhất trong dãy đó. In ra số lượng phần tử của dãy con. Các số trong phạm vi longint.

Input:

- Dòng đầu tiên gồm số nguyên N.

- Dòng thứ hai gồm N số mô tả dãy.

*Output:* Gồm một số nguyên duy nhất là đáp số của bài toán

*Ví dụ:*

Input

5

2 1 4 3 5

Output

3

*Gợi ý giải:*

- Sử dụng phương pháp chặt nhị phân.

- Gọi Min[i] là giá trị nhỏ nhất của phần tử cuối cùng trong dãy con tăng dài nhất có độ dài là i. Ta khởi tạo Min[0] = .

Gọi l là độ dài dãy con tăng dài nhất hiện tại. Ta nhận thấy Min[0..l] là mảng tăng dần. Với mỗi phần tử a[i] trong mảng, có 2 khả năng xảy ra:

+ a[i] > Min[l], khi đó Min[l+1] = a[i] và độ dài dãy con tăng dài nhất tăng lên 1.

+ Min[k-1] < a[i] Min[k], khi đó nếu a[i] < Min[k] thì ta cập nhật lại Min[k] = a[i]. Vì Min là mảng tăng dần nên ta có thể tìm ra k bằng chặt nhị phân.

- Độ phức tạp: nlog(n).