**ĐỀ NGÀY 19 – 1 - 2022**

**Bài 1. KSTR**

Cho số nguyên dương và tập hợp khác rỗng . Tập gồm các phần tử khác nhau

Người ta định nghĩa phép toán là những phần tử chỉ thuộc tập và không thuộc tập

**Ví dụ**: và khi đó

Dễ dàng nhận thấy phép toán trên không có tính kết hợp, tức là nên chúng ta quy ước thứ tự thực hiện phép toán là thực hiện từ phải qua trái.

**Ví dụ**:

**Yêu cầu**: Hãy xác định số cách chọn các tập từ tập sao cho được kết quả là tập có ít nhất phần tử khác nhau.

**Dữ liệu:** Vào từ file văn bản **KSTR.INP**

* Dòng đầu tiên chứa hai số nguyên dương và
* dòng tiếp theo, dòng thứ mô tả tập chứa các số trong đó là số lượng phần tử của tập , là các phần tử của tập .

*Các số trên một dòng của input file được ghi cách nhau bởi dấu cách.*

**Kết quả:** Ghi ra file văn bản **KSTR.OUT** một số nguyên duy nhất là kết quả bài toán (*lấy theo modulo* ***123457***)

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **KSTR.INP** | **KSTR.OUT** | **Giải thích** |
| 3 3  5 5 6 7 8 9  3 4 5 6  3 7 8 9 | 6 | Có thể chọn: |

**Bài 2. Tìm đường đi trên lưới**

Bạn đang được giao việc viết module tìm đường trong một hệ thống trí tuệ nhân tạo (AI). Công việc của bạn là tìm đường ngắn nhất giữa hai điểm tọa độ trên một mặt phẳng. Vị trí xuất phát có tọa độ (0, 0), vị trí cần tìm đường có tọa độ (Ex, Ey). Tuy vậy, công việc không chỉ đơn giản là xuất ra |Ex| + |Ey| do một số quy luật đặc biệt.

Tại một vị trí (x, y), bạn chỉ được di chuyển theo 2 hướng:

* Nếu x lẻ bạn được di chuyển đến vị trí (x, y – 1).
* Nếu x chẵn bạn được di chuyển đến vị trí (x, y + 1).
* Nếu y lẻ bạn được di chuyển đến vị trí (x - 1, y).
* Nếu y chẵn bạn được di chuyển đến vị trí (x + 1, y).

Dữ liệu vào được cho trong file POGRID.IN:

* Dòng thứ nhất ghi số T là số test.
* T dòng tiếp theo, mỗi dòng ghi 2 số Ex, Ey.

Dữ liệu ra là file POGRID.OUT:

* Với mỗi dòng trong input, in ra một số là độ dài đường đi ngắn nhất.

Ví dụ:

|  |  |
| --- | --- |
| POGRID.IN | POGRID.OUT |
| 2  0 -4  5 4 | 8  9 |

Giới hạn:

* Có 2 test, trong mỗi test, T ≤ 100.
* Trong test 1, |Ex| ≤ 100, |Ey| ≤ 100.
* Trong test 2, |Ex| ≤ 106, |Ey| ≤ 106.

**Bài 3. ROBOT**

HD vừa sáng tạo ra một trò chơi điều khiển robot mới cho 2 bé Bi, Bo chơi. Nội dung trò chơi như sau:

* Có N cây cột đánh số từ 1 đến N, cây cột thứ có chiều cao

h

h-t

* Có M đường nhảy dạng tương ứng là nhảy từ cây sang cây (hoặc từ cây j sang cây i) mất và nếu nhảy từ độ cao của cây thì sang cây sẽ có độ cao là với điều kiện
* Nếu robot di chuyển lên xuống trên cột hiện tại, thời gian di chuyển mất trên di chuyển.

Hiện tại robot đang ở độ cao X của cây 1, Bi-Bo cần phải tìm phương án di chuyển nhanh nhất đến độ cao của cây N. Bạn hãy giúp 2 bé Bi-Bo tính thời gian di chuyển ngắn nhất thỏa mãn yêu cầu đầu bài?

**Dữ liệu:** Vào từ file văn bản **ROBOT.INP**

* Dòng 1: Chứa 3 số nguyên dương N, M, X tương ứng là số lượng cây cột, số lượng đường nhảy và độ cao của robot đang ở cột 1.
* N dòng tiếp theo, dòng thứ chứa 1 số nguyên dương tương ứng là chiều cao của cột .
* M dòng tiếp theo, mỗi dòng chứa 3 số nguyên dương tương ứng là nhảy từ cây sang cây (hoặc từ cây sang cây ) mất

*Các số trên một dòng của input file được ghi cách nhau bởi dấu cách.*

**Kết quả:** Ghi ra file văn bản **ROBOT.OUT** một số duy nhất là thời gian ngắn nhất để robot di chuyển đến độ cao của cây N, nếu không thể di chuyển đến thì ghi -1

**Ví dụ:**

|  |  |  |
| --- | --- | --- |
| **ROBOT.INP** | **ROBOT.OUT** | **Giải thích** |
| 5 5 0  50  100  25  30  10  1 2 10  2 5 50  2 4 20  4 3 1  5 4 20 | 110 | Trèo lên 50(m) ở cây 1 mất 50(s)  Nhảy từ cây 1 sang cây 2:   * Mất 10(s) * ở độ cao 40 trên cây 2   Nhảy từ cây 2 sang cây 4:   * Mất 20(s) * ở độ cao 20 trên cây 4   Nhảy từ cây 4 sang cây 5:   * Mất 20(s) * ở độ cao 0 * trèo thêm 10(m) mất 10(s)   Tổng thời gian: 110(s) |
| **ROBOT.INP** | **ROBOT.OUT** | **Giải thích** |
| 2 1 0  1  1  1 2 100 | -1 | Từ cây 1, bất kỳ độ cao nào, khi nhẩy sang cây 2 đều không thực hiện được vì |
| **ROBOT.INP** | **ROBOT.OUT** | **Giải thích** |
| 4 3 30  50  10  20  50  1 2 10  2 3 10  3 4 10 | 100 | Di chuyển xuống 10(m) ở cây 1 mất 10 (s) và đang ở độ cao 20(m)  Nhảy sang cây 2:   * Mất 10(s) * ở độ cao 10 trên cây 2.   Nhảy sang cây 3:   * Mất 10(s) * Ở độ cao 0(m), trèo lên 10(m) mất 10(s), ở độ cao 10(m);   Nhảy sang cây 4:   * Mất 10(s), * ở độ cao 0 (m), trèo lên 50(m) mất 50(s)   Tổng thời gian: 100(s) |

**Chú ý:**

* *25% số điểm tương ứng với các test có:*
* *25% số điểm tương ứng với các test có*