Towers of Hanoi Assignment

DUE: November 12, 2019

Write the functions specified below to find the best-known solutions to the Towers of Hanoi problem when the # of towers may be greater than 3. Your functions should handle between 3 and 10 towers. and the number of disks should be at most 10000, but you may assume that it is small enough to ensure that the number of moves fits in a signed 8-byte long.

The initialization function should initialize the arrays **n_hanoi** and **k_hanoi**, where **n_hanoi[twr][dsks]** should be the minimum # of moves to solve the problem with the # of towers and disks being **twr** and **dsks**, respectively, and the **k_hanoi** array should give the corresponding # of disks **k** that should be moved to a spare tower first, to achieve the value in **n_hanoi.**

Recall that the solution for 3 towers is most easily implemented as a function

```
void hanoi(int n, int from, int to, int spare);
```

For your more complicated version, the argument aux serves the same purpose as the 3 arguments from, to, spare.

Suggestion: Use aux[0] as the start tower, aux[1] as the destination tower, and aux[n_twrs-1] as the first spare tower.

```
typedef vector<int> VI;
typedef vector<long> VL;
vector<VL> n_hanoi;
vector<VI> k_hanoi;

// Initialize n_hanoi and k_hanoi
void hanoi_init();

// Fill moves so that each element is a two-integer VI describing the move
// You may assume that, initially, aux[] = {0, 1, 2, ..., n_disks - 1}
void hanoi(vector<VI>& moves, int n_twrs, int n_dsks, VI& aux);
```