BCH 571 Bioinformatics for Life Scientists

Lab 7

Program 1- (call this lab7. py)

Implement the HanoiTowers function in Python (provided in pseudocode below). Benchmark the time cost of running from 1 disk up to 23 disks by calling the function for each number of disks from 1 to 23, inclusive. Remember to make the first call with fromPeg=1 and toPeg=3. Tabulate the time cost and make a plot with your favorite plotting program (excel, etc.). How does the program scale?

To determine the time cost of a function call, you will want to use the function (from module time) called time(). That returns the number of seconds from the epoch (Jan 1, 1970). You can call time.time() before and after a function call. The difference is the time elapsed during the function call.

```
HanoiTowers(n,fromPeg,toPeg)
  if n=1
    output "Move disk from peg fromPeg to peg toPeg"
    return
  unusedPeg ← 6 – fromPeg – toPeg
  HanoiTowers(n-1,fromPeg,unusedPeg)
  output "Move disk from peg fromPeg to peg toPeg"
  HanoiTowers(n-1,unusedPeg,toPeg)
  return
```

Part 2 (Not a program!)

In class, I provided the numerically stable method for solving the quadratic equation. The equations are below. Translate this procedure into pseudocode.

$$X1 = \frac{-b - \text{sign}(b)\sqrt{b^2 - 4ac}}{2a}$$

$$x2 = c\ /\ (a\ x1)$$

Note that square root is a general mathematical expression, i.e. it is acceptable in pseudocode. sign() is not standard, and should be described in the pseudocode.