J. App. Em. Sc: 1(1) 101-106, June 2004

From Silicon to Bio-molecular Computing

M. Nawaz
nawaz@buitms.edu.pk

Ikramullah Khan
ikram@buitms.edu.pk

Balochistan University of Information Technology and Management Sciences
Jinnah Town Quetta.

Abstract:

Several problems from mathematics and computer science e.g. Harmonic Series, Tower of Hanoi and Traveling Salesman Problem require huge amount of memory and lot of computational power. Several attempts have been made to solve the TSP, however no silicon chip based solution has been found so far. This famous problem however has been solved by Leonard Adleman using DNA computation. Our paper calls for the attention of Bio-Information experts and Computer Scientists for making a model of computation for such problems.

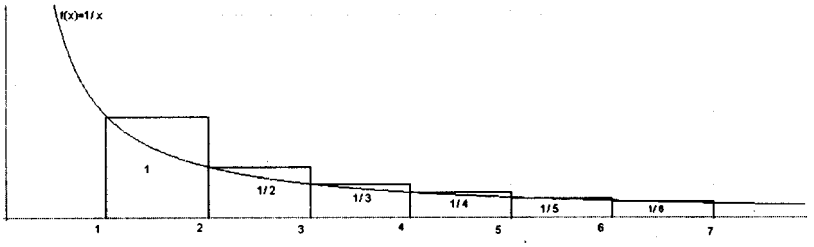
First we discuss few famous problems from Mathematics and Computer Science and algorithms for their solutions. Finally we will comment on DNA computing.

Harmonic Series

The series

$$1+\frac{1}{2}+\frac{1}{3}+\ldots$$

is called the harmonic series. Books on calculus contain theoretical proofs for divergence of this series. A simple argument to this effect is as follows.

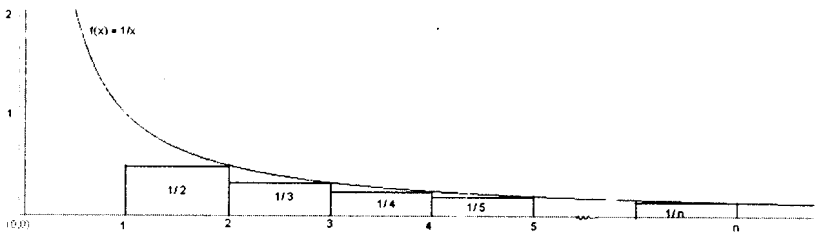


In the graph drawn above, the harmonic series is given by the sum of the areas of the rectangles. Since this area is larger than the area under the graph, we have:

$$1+\frac{1}{2}+\frac{1}{3}+\ldots > \int_1^\infty \frac{1}{x}\,dx = \ln x|_1^\infty = \infty$$

From this we conclude that the sum of the areas of the rectangles is infinite and hence the harmonic series diverges.

However there is no empirical evidence that this series is divergent. For instance suppose that someone starts adding terms of this series at the time when this universe came into being that is thirteen billion (13×10^9) years ago, and adds each of the next terms after every microsecond (that is 1000000 terms per second) then by now this series would have yielded just 55.36!! For proof consider the following graph



This graph shows that

$$H_n < 1 + \int_1^n (1/x)\, dx.$$

And therefore

$$H_n < 1 + \ln n. \text{ -------- (a)}$$

let

$$n = 13 \times 10^9 \times 365 \times 24 \times 60 \times 60 \times 10^6$$

then

$$\begin{aligned} \ln n &= \ln 13 + \ln 10^9 + \ln 365 + \ln 24 + \ln 60 + \ln 60 + \ln 10^6 \\ &= 2.56 + 20.72 + 5.90 + 3.18 + 4.09 + 4.09 + 13.82 \\ &= 54.36 \end{aligned}$$

Therefore from (a) we have

$$H_n = 1 + 1/2 + 1/3 + 1/4 + \ldots + 1/n < 55.36$$

Divergence of harmonic series implies that for any given real number say Υ, we can find a natural number N such that $H_n > \Upsilon$ whenever $n > N$. For $\Upsilon = 1000$, imagine that how long will it take (@ 1000000 terms per second) to sum the harmonic series over to this value of Υ? Harmonic series, being a divergent series, must sum up to this number theoretically. And what if Υ is even larger.

The Tower of Hanoi

An ancient tale tells about a king who received a service from a wise man and insisted on repaying the favour. The wise man asked for a payment of rice for each square of the king's chessboard on whose first square is placed a single grain of rice, on whose second square are placed two grains, four on the third square, and so on. The king, proudly granted the wish, at least until it became clear that it will wipe out all his stock (the last square requires 2^{63} grains of rice which weighs over 3750 billion maunds! Calculations are based upon: 8 rice grains = 1 ratti, 8 rattis = 1 masha, 12 mashas = 1 tola, 5 tolas = 1 chhatank, 16 chhatanks = 1 seer, 40 seers = 1 maund)

In another similar instance we are given a tower (called the tower of Hanoi) of eight disks, initially stacked in decreasing size on one of three pegs. The objective is to transfer the entire tower to one of the other

pegs, moving only one disk at a time and never a larger one onto a smaller. The figure below shows the initial state of discs on the source peg.

It can be proved that $2^n - 1$ moves are required to transfer n disks to one of the other pegs subject to the conditions given above [3]. Thus it will take 255 moves to transfer all the 8 disks. For n = 64, the number of moves required is $2^{64} = 18446744073709551615$. It will take 5000 centuries to shift the stack of 64 disks, at the rate of 1000000 moves per second!

Edouard Lucas [1], a French mathematician furnished a toy with a romantic legend about Tower of Brahma, which supposedly has 64 disks of pure gold resting on three diamond needles. At the beginning of time, he said, God placed these golden disks on the first needle and ordained that a group of priests should transfer them to the third, according to the rules above. The priests reportedly work day and night at their task. When they finish, the Tower will crumble and the world will end.

The question here is that can we define some computing model based on the DNA for the above two famous problems of mathematics and computer science, namely the harmonic series and the tower of Hanoi.

Harmonic Series (revisited)

We have seen that

$$1 + \frac{1}{2} + \frac{1}{3} + \ldots$$

is a very slow growing sum. The mathematical proof given above is very sound. But a computer scientist would really appreciate this proof much more if he can develop a program that actually does the addition of each and every term of the above series up to the number of seconds in 13 billion years. The limiting capabilities of the silicon computer, however, will never give the result in real time, as we have shown below. Hence the need for an unconventional model of computing can be appreciated with the help of the difficulties we are confronted with in proving the famous mathematical theorems using programming.

We have constructed two very simple algorithms that demonstrate the above fact. Following are two separate algorithms that will try to add up every element of the above series up to the following number, which is the number of seconds in 13 billion years. We call this number SEC_IN_THIR_BILL_YRS;

SEC_IN_THIR_BILL_YRS = $13 * 365 * 24 * 60 * 60 * 10^9$

That is to say that in 13 billion years, each year has 365 days, each day has 24 hours, each hour has 60 minutes and each minute has 60 seconds, thus the number above becomes.
SEC_IN_THIR_BILL_YRS = $409968 * 10^{12}$

Algorithm1 (Iterative way):
procedure HarmonicSum(SEC_IN_THIR_BILL_YRS: Integer): float
begin

```
        result := 0.0
        for i :=1 to SEC_IN_THIR_BILL_YRS
        begin
          result := result+(1/i)
        end
        return result
        {the variable 'result' holds the final value of the summation}
end
```

We tested the above program on ANSI C compiler and Pentium 4 processor with a clock speed of 2.7GHz on Windows XP. After 30 minutes of execution, the program summed only up to 19. Since every next element in the series is smaller than the previous one, getting every next integer for example will take more time than the previous integer. The algorithm above is written in pseudo-code not in the real code. The output of the program may not give the exact value above; this value depends upon the language the real code was written in, the compiler's accuracy of decimal point values and the underlying operating environment.

Here is another version of the above program that is simply its numerical equivalent.

Algorithm2 (Numerical way:)

```
procedure HarmonicSum(SEC_IN_THIR_BILL_YRS: Integer): float
begin
    result
    x1 :=ln(13)
    x2 :=ln(365)
    x3 :=ln(24)
    x4 :=ln(60)
    x5 :=ln(10)
    result = 1+x1+x2+x3+x4+x4+9*x5 + 6*x5
    return result
   {the variable 'result' holds the final value of the summation}
end
```

Output : The natural log is 55.36

The Tower of Hanoi (revisited)

The puzzle is well known to students of Computer Science since it appears in virtually any introductory text on data structures or algorithms. Its solution touches on two important topics discussed later on:

- recursive functions and stacks
- recurrence relations

The procedure for playing with the discs can generally be drawn as.

1. Move the top N-1 disks from Src to Middle peg(using Dst as an intermediary peg)
2. Move the bottom disks from Src to Dst

3. Move N-1 disks from Middle peg to Dst peg(using Src as an intermediary peg)

Therefore an equivalent algorithm for the procedure above can easily be constructed as follows

Algorithm3:

procedure TowerOfHanoi (N, Src, Aux, Dst :Integer):Integer

begin

if N is 0 exit
TowerOfHanoi(N-1, Src, Dst, Aux)
Move the disc from Src to Dst
TowerOfHanoi(N-1, Aux, Src, Dst)

end

Statistics that we obtained using our programs are shown below

Number of Discs	*Number of Moves required*
1	*1*
2	*3*
3	*7*
4	*15*
5	*31*
6	*63*
7	*127*
8	*255*
...	...
...	...
64	*18446744073709551616*

This was just an example of a mathematics puzzle. But the recurrence relation of the system above is very useful in many other applications of computer science and mathematics. Note the final number of moves if the discs are increased to 64. Now, if a machine can do 1000 moves per second, it will still take approximately 584942417 years to move all the discs from first peg to the last one.

A breakthrough: Can we go forward

Leonard Adleman [2] in 1994 made a breakthrough in the field of nontraditional computing by solving the famous computer science's Traveling Salesman Problem (TSP for short) using DNA as a computing substrate. He successively represented each city with a DNA strand of twenty molecules. If two cities had a link between them, it was represented by another third DNA strand used as a link, using the rules of DNA molecule-combination. He literally connected all the cities in a small test tube. The number of cities was kept large enough to make the problem complex. But every one was shocked when he finally got the smallest path from the start to the end such that it passed

through all the cities only once. He got this by continuously neglecting many unwanted paths using the graph theory concepts on the biological establishment of the DNA in the test tube. A rough representation of two cities 1 and 2 linked together is shown in the diagram below.

Now the question is that can we present biological solutions to the problems that we have mentioned above? No doubt there are limitations of the DNA when it comes to computing, but this is the area where the research is headed to. Computer scientists and Bio-informatists are struggling hard to overcome these limitations.

References

1. Edouard Lucas, Recreations Mathematiques volume 3, Gauthier-Villars, Paris 1891-1894. Reprinted Albert Blanchard, Paris 1960.
2. Leonard M. Adleman, Computing with DNA, Scientific American, August, 1998, 54-61.
3. Ronald L. Graham, Donald E. Knuth, Oren Patashnik, Concrete Mathematics, Sixth printing, 1990, Addison-Wesley Publishing Company.