CSCI 3104 Summer 2019 Problem Set 1
Ian Peterson (03/20) E-mail: Ian.Peterson@Colorado.edu

Compiled with LaTeX

1. *Problem 1*

   Give an example of an application that uses a proprietary algorithm (i.e. Spotify's "Discover Weekly" playlist, Google's PageRank algorithm, etc.). Find an article that discusses this algorithm and give a summary of its content. Provide at least 4-5 sentences for full credit.

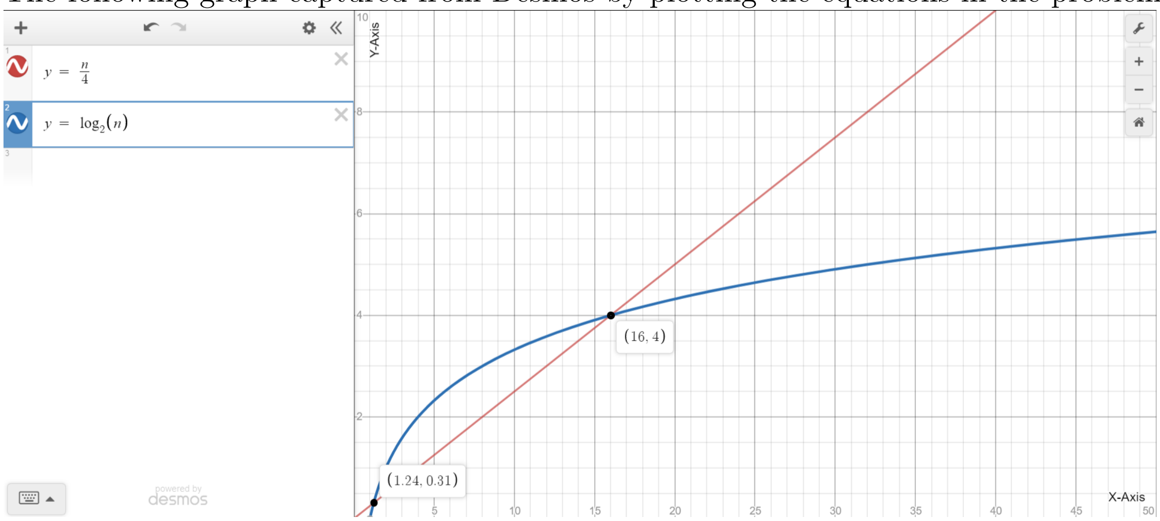
   Answer:

   https://steamcommunity.com/games/593110/announcements/detail/1301948399257707760
   This article posted by valve reveals the ways which the steam store works, and a multitude of defences for this way of working. Due to it's many intricacies, i will keep a description on a high level. It first tries to categorize each user into a number of groups based upon factors such as their play history (whether they just play AAA titles, or search for gems in the many indie games available), their visit frequency (whether they browse the shop daily of monthly), their common attributes (what tags the games have that they buy/play), and a number of other things. It then attempts to recommend games to you based on the following categories in descending order: Similarity to games you've played, User Reviews of these titles, Recommendations by followed curators, and users on your friend's list who own/wish listed this game. After compiling a list of titles the algorithm thinks you'd like, it tries to sort the first few elements of the list according to your recent trends in playing or viewing, meaning if recently you have been playing a lot of strategy games, the strategy titles will appear to you first. Finally, it will group similar titles together because the steam store recommended display in groups of a few tiles, so the algorithm will find a few titles with common traits near each other on the list, and display them together with a title showing why they recommend these games to you. A complex algorithm such as this one is critical for an application with so many products and users, that way users can find the things they want, and all developers can reach their desired audience.

2. *Problem 2*

Consider two algorithms that perform the same function, that run in $n/4$ and $log_2(n)$, respectively, where $n \in \mathbb{N}$ (i.e. natural numbers). Plot these run times on the same graph with the values $n \in [1, 50]$ (don't forget labels). Provide the set of intervals over $\mathbb{N}$, where $n/4$ is the strictly better algorithm to use (think greater than, not greater than or equal).

Answer:
The following graph captured from Desmos by plotting the equations in the problem.



The algorithm what runs in $n/4$ is better when it is strictly below, and not equal to the line for $log_2(n)$ because at any point on the X-axis, representing N where $n/4$ has a smaller value compared to $log_2(n)$, it must be running in less time, thus is better, or more efficient. This occurs on the interval $n \in (1.24, 16)$ (note non-inclusive) which is apparent when evaluating the graph above for the two plotted equations.

3. *Problem 3*

Harry the Wizard needs your help solving a riddle deep in an abandoned dwarven mine. There are two doors marked A and B, respectively, and a stone pedestal in the middle of the room inscribed with the following text:

"The dwarves who dwelled in this mine were fond of mathematical drinking games. Two dwarves, Arnold and Barry, are chosen as the participants for this game, and pick functions that they think will best predict the number of people who enter the pub in an hour (p) based on the number of drinks they consume in that hour (d). They choose $p(d) = d/2$ and $p(d) = 2ln(d)$, respectively. Below is a record of the number of drinks consumed and the corresponding number of patrons patrons who entered the bar over four hours.

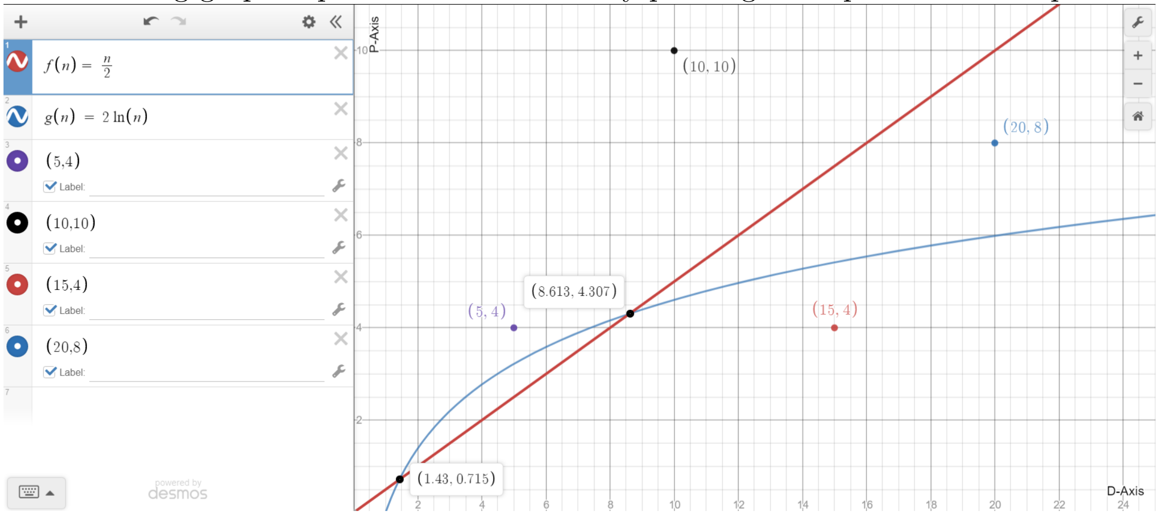
Number Drinks (d)	Number Patrons (p)
5	4
10	10
15	4
20	8

Which dwarf, Arnold or Barry, chose the most accurate function?"

Due to an error in Harry's mental calculations in the last puzzle causing Grog the Barbarian to lose his pinky finger, the party demands a written explanation of the solution to the puzzle. Additionally, since Grog doesn't know how to read, provide a relevant figure in your solution so Grog can believe he is part of the discussion.

Answer:

The following graph captured from Desmos by plotting the equations in the problem.



Above is plotted the two estimate equations as well as the four points presented in the table given in the problem. At first glace it appears that the linear equation is a better approximation, but to better estimate which is better, I will calculate the total of the differences for the two lines to see who had a better estimate. The line $p(d) = d/2$ has a total difference of exactly 12, while the curve $p(d) = 2ln(d)$ had a total difference of 9.601 across all the points, which means my instinct was wrong and the $p(d) = 2ln(d)$ curve is the more accurate one, so Barry must have been the correct dwarf, and this furthermore implies that Harry should go through door B, corresponding with Barry the dwarf.

4. *Problem 4*

Consider the following recurrence relation:

$$G_n = \begin{cases} 1 & \text{if } n = 0 \\ \text{-1} & \text{if } n = 1 \\ 2 & \text{if } n = 2 \\ (G_{n-1})(G_{n-2}){+}G_{n-3} & \text{otherwise} \end{cases}$$

(a) *subproblem a*

Write pseudo code for this function that takes in a positive integer, n, and returns the nth number in the sequence.

Answer:

```
int foo(n)//n is a positive integer as stated in the problem.
    new array A[1,-1,2] //new array to store the known values
    for i from 3 to n-1 //n-1 used because indexing at zero
        A.append(A[i-1]*A[i-2]+A[i-3]) //add new element
    return A[n-1] //return desired element
```

(Proof of algorithm found in subproblem b)

(b) *subproblem b*

What is the 10th number in the sequence?

Answer:

Plugging a 10 into this pseudo code goes through a loop, and winds up providing the answer -147,503,198. I found this by manually calculating the elements in the array. As follows, $foo(0) = 1; foo(1) = -1; foo(2) = 2; foo(3) = -1; foo(4) = -3; foo(5) = 5; foo(6) = -16; foo(7) = -83; foo(8) = 1333; foo(9) = 110655; foo(10) = -147503198$. To prove this is true we must do a Loop Invariant Proof, found on the following page.

L.I.
At the start of each iteration, the sub array A[0...i-1] contains the first i elements in the sequence described by the problem.
Init.(i = 3)
In the sub array A[0...2], The elements that exist are only the three presented in the problem, so the L.I. is maintained trivially.
Maint.(i = k)
Assume our L.I. holds for $i = k-1$. We want to append a new element to our array of correct elements by following the rule presented in the problem description, being $A[i] = A[i-1] * A[i-2] + A[i-3]$ by line 4. We know the added value is correct so then the i variable increments, and we can again for i = k+1 claim that the array is correct up to k rather than up to k-1.
Term.(i = n)
When i = n, the sub array A[0...n-1] now contains the nth element at the index n-1 because we began our indexing at 0, so we return A[n-1], which because it will accurately portray the nth element of the sequence, proves that our algorithm is correct.