

FiveThirtyEight's September 18, 2020 Riddler

Emma Knight

September 20, 2020

This week's riddler is about hunting for words:

Question 1. *One of Ollie's favorite online games is Guess My Word. Each day, there is a secret word, and you try to guess it as efficiently as possible by typing in other words. After each guess, you are told whether the secret word is alphabetically before or after your guess. The game stops and congratulates you when you have guessed the secret word. For example, the secret word was recently "nuance," which Ollie arrived at with the following series of nine guesses: naan, vacuum, rabbi, papa, oasis, nuclear, nix, noxious, nuance.*

Each secret word is randomly chosen from a dictionary with exactly 267,751 entries. If you have this dictionary memorized, and play the game as efficiently as possible, how many guesses should you expect to make to guess the secret word?

Define $f(n)$ to be the expected number of guesses it takes with n words. Since the optimal strategy is clearly to guess the middle most word, which splits the set of words into the two smallest possible sets. One gets the following recursion for $f(n)$:

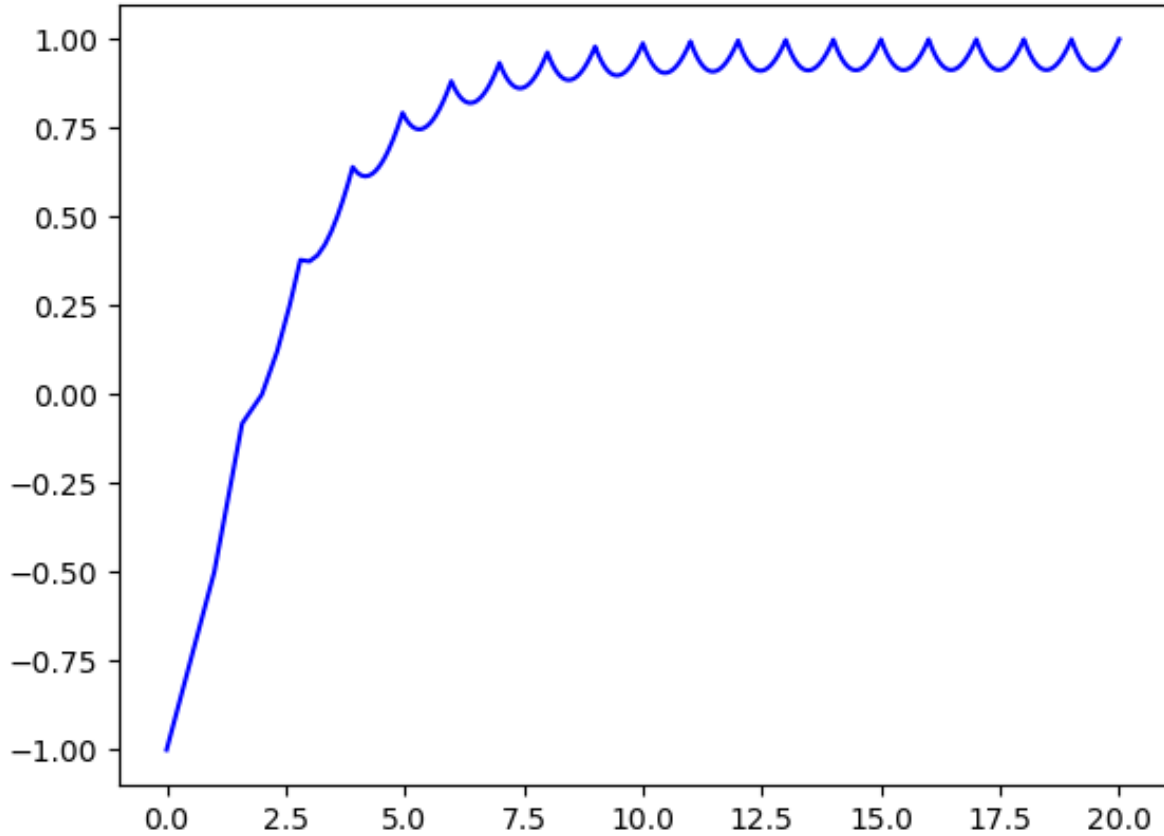
- $f(1) = 1$ and $f(2) = \frac{3}{2}$
- $f(2k + 1) = 1 + \frac{2k}{2k+1}f(k)$
- $f(2k) = 1 + \frac{1}{2}f(k) + \frac{k-1}{2k}f(k-1)$

This is a little messy, but notice that if you define $g(n) = nf(n)$, then g admits a nice recursion:

- $g(1) = 1$ and $g(2) = 3$
- $g(2k + 1) = 2k + 1 + 2g(k)$
- $g(2k) = 2k + g(k) + g(k - 1)$

This is something that is very easy to quickly code up on a computer, and one gets that $g(267751) = 4563001$, yielding $f(267751) = \frac{4563001}{267751} = 17.041956\dots$

A natural question is “How big is $f(n)$?” One can see that $f(n) \approx 1 + f(\frac{n}{2})$, so it is natural to expect that $f(n) \approx \log_2(n)$. To get a better sense of how big $f(n)$ is, I computed $\log_2(n) - f(n)$ for n between 1 and $2^{20} - 1$, plotting $\log_2(n) - f(n)$ on the y -axis and $\log_2(n)$ on the x -axis:



One can see that very quickly one has that $\log_2(n) - f(n)$ starts oscillating between .91 and 1, being very close to 1 when n is close to a power of 2 and close to .91 when n is roughly $\sqrt{2}$ times a power of 2. One can compute $f(2^n - 1)$ as follows:

$$\begin{aligned}
 g(2^n - 1) &= 2^n - 1 + 2g(2^{n-1} - 1) \\
 &= (2^n - 1) + 2(2^{n-1} - 1 + 2g(2^{n-2} - 1)) \\
 &= (2^n - 1) + (2^n - 2) + 4g(2^{n-2} - 1) \\
 &\vdots \\
 &= (2^n - 1) + (2^n - 2) + \dots + (2^n - 2^{n-1}) \\
 &= n2^n - (2^n - 1) \\
 &= (n - 1)(2^n - 1) + n
 \end{aligned}$$

giving $f(2^n - 1) = n - 1 + \frac{n}{2^n - 1}$. One can similarly get $f(2^n) = n - 1 + \frac{n+2}{2^n}$ to get that $\log_2(2^n) -$

$f(2^n) = 1 - \frac{n+2}{2^n}$, which tends to 1 very quickly. I don't have as good of a way to see where the troughs are. I think it's still reasonable to say that, for $n \gg 0$, $\log_2(n) - 1 < f(n) < \log_2(n) - .9$.