PORTER'S CONSTANT - EXPOSITORY

SATVIK BALAKRISHNAN

Contents

1. Overview 1
2. Introductory Definitions 2
3. Porter's Constant Derivation 3
3.1. Counting Relatively Prime Pairs 3
3.2. Asymptotic Analysis 3
3.3. Logarithmic Approximation 4
3.4. Constant Factor 4
3.5. Final Result 5
References 5

1. Overview

In mathematics, Porter's constant is a constant that arises in the study of the efficiency of the Euclidean Algorithm. Porter's constant, named after its discoverer, Professor John Porter, has significant implications in algorithmic efficiency analysis. Understanding its properties can lead to insights into the behavior of the Euclidean Algorithm and related computational problems. In this paper we will cover what porter's constant is, do our best to derive it, and cover some applications of porter's constant.

Date: February 2024.

2. Introductory Definitions

Before we get into understanding porter's constant, we must first understand what the euclidean algorithm is which brings us to our first definition.

Definition 2.1. (Euclidean Algorithm) The Euclidean algorithm provides an algorithm for finding the largest common divisor of two numbers a and b.

(1) Given two integers a and b where $a > b$, compute the quotient q and remainder r from the division of a by b:

$$a = bq + r, \quad 0 \leq r < b$$

(2) Replace a with b and b with r.
(3) Repeat step 1 until $r = 0$.
(4) The GCD of a and b is the last non-zero remainder.

Proof. Let $d = \gcd(a, b)$. By definition, d divides both a and b. We can write a as

$$a = bq + r,$$

where q is the quotient and $r = a \mod b$ is the remainder when a is divided by b.

Since d divides both a and b, it must also divide their linear combination. Thus, d divides r, because

$$r = a - bq.$$

Therefore, d is a common divisor of b and r.

Let d' be any common divisor of b and r. Then d' divides b and r, and hence d' divides a, because

$$a = bq + r.$$

Therefore, d' is a common divisor of a and b.

It follows that the set of common divisors of a and b is the same as the set of common divisors of b and r. Thus, the greatest common divisor of a and b is the same as the greatest common divisor of b and r:

$$\gcd(a, b) = \gcd(b, a \mod b).$$

■

We now use this lemma to prove the correctness of the Euclidean algorithm.

Let $a_0 = a$ and $b_0 = b$. Consider the sequence of pairs (a_i, b_i) generated by the Euclidean algorithm. By the lemma, we have

$$\gcd(a_i, b_i) = \gcd(a_{i+1}, b_{i+1}),$$

for all i.

The algorithm terminates when $b_i = 0$, and at that point, a_i is the greatest common divisor. Hence, the Euclidean algorithm correctly computes the gcd of a and b.

Porter's constant shows up in the study of the efficiency of this algorithm. Before deriving Porter's constant we need to be familiar with a few more functions and constants.

Definition 2.2. (Euler-Mascheroni constant) The Euler-Mascheroni constant, denoted as γ, is defined as the limiting difference between the harmonic series and the natural logarithm. Mathematically, it is expressed as follows:

$$\gamma = \lim_{n\to\infty}\left(\sum_{k=1}^{n}\frac{1}{k} - \ln n\right).$$

The constant γ approximately equals 0.57721.

Definition 2.3. (Glashier-Kinkelin constant) The Glaisher-Kinkelin constant, denoted as A, is a mathematical constant related to the product of powers of integers. It is defined by the limit:

$$A = \lim_{n\to\infty}\frac{\left(\prod_{k=1}^{n}k^k\right)}{n^{n^2/2+n/2+1/12}e^{-n^2/4}}.$$

Definition 2.4. (Riemann-Zeta function) The Riemann zeta function, denoted as $\zeta(s)$, is a function of a complex variable s defined for $\mathrm{Re}(s) > 1$ by the infinite series:

$$\zeta(s) = \sum_{n=1}^{\infty}\frac{1}{n^s}.$$

Definition 2.5. ($T(n)$) We let the function $T(n)$ be the average number of steps taken by Euclid's Algorithm. Mathematically:

$$T(n) = \frac{1}{\phi(n)}\sum_{\substack{1\le m<n\\ \gcd(m,n)=1}}T(m,n).$$

where $\phi(n)$ is the totient function.

Definition 2.6. (The Totient Function) $\phi(n)$ is Euler's totient function, which represents the number of integers up to n that are relatively prime to n.

Now that we know what the Euclidean Algorithm, Euler-Mascheroni constant, Glashier-Kinkelin constant, and Riemann-Zeta function are, we can get into deriving Porter's constant.

3. Porter's Constant Derivation

We first need to compute the expected number of steps $E(T(n))$ for Euclid's algorithm over all pairs of relatively prime integers (m, n) with $1 \le m < n$.

3.1. Counting Relatively Prime Pairs.

The number of pairs (m, n) where $\gcd(m, n) = 1$ and $1 \le m < n$ can be determined using Euler's totient function $\phi(n)$.

Euler's totient function $\phi(n)$ counts the number of integers up to n that are relatively prime to n. For a large n, the probability that two randomly chosen integers are relatively prime is given by:

$$P(\gcd(m,n) = 1) = \frac{6}{\pi^2}$$

This is based on the fact that the sum of the reciprocals of the squares of positive integers converges to $\frac{\pi^2}{6}$.

3.2. Asymptotic Analysis.

Sum over all pairs of relatively prime integers and convert the sum into an integral.

3.2.1. *Summing Over Steps.* Let $T(m, n)$ be the number of steps Euclid's algorithm takes for input (m, n). We want to compute the average number of steps:

$$E(T(n)) = \frac{1}{\phi(n)} \sum_{\substack{1 \le m < n \\ \gcd(m,n)=1}} T(m, n)$$

For large n, we can approximate this sum by considering all pairs (m, n) and then adjusting by the probability of being relatively prime:

$$E(T(n)) \approx \frac{1}{n-1} \sum_{m=1}^{n-1} T(m, n)$$

3.3. **Logarithmic Approximation.** Use properties of logarithms to identify the leading term in the summation.

3.3.1. *Harmonic Series and Logarithms.* The average number of steps in Euclid's algorithm is related to the harmonic series. The harmonic series H_n is:

$$H_n = \sum_{k=1}^{n} \frac{1}{k}$$

For large n, H_n can be approximated by:

$$H_n \approx \ln n + \gamma$$

where γ is the Euler-Mascheroni constant.

3.3.2. *Converting Summation to Integral.* To approximate the summation, convert it into an integral:

$$\sum_{m=1}^{n-1} \frac{1}{m} \approx \int_1^n \frac{1}{x}\, dx = \ln n$$

Thus, the average number of steps can be approximated as follows.

$$E(T(n)) \approx \frac{1}{n} \int_1^n T(x, n)\, dx$$

3.4. **Constant Factor.** Determine the exact constant factor $\frac{12 \ln 2}{\pi^2}$ by a detailed calculation.

3.4.1. *Detailed Calculation of the Integral.* Using results from number theory and the asymptotic analysis, it has been shown that the average number of steps can be precisely calculated. The key result involves integrating over relatively prime pairs and using known constants:

$$\int_1^n \frac{T(x, n)}{n}\, dx \approx \frac{12 \ln 2}{\pi^2} \ln n$$

This result comes from detailed number-theoretic analysis, specifically from work by Heilbronn.

3.5. **Final Result.** The expected number of steps $E(T(n))$ for Euclid's algorithm, averaged over all relatively prime pairs (m, n) with $1 \leq m < n$, is given by:

$$E(T(n)) \approx \frac{12 \ln 2}{\pi^2} \ln n + O(1)$$

This completes the derivation of Porter's constant.

References

J. W. Porter, *On a theorem of Heilbronn*, Mathematika, vol. 22, pp. 20–28, 1975.

Wikipedia contributors, *Porter's constant*, `https://en.wikipedia.org/wiki/Porter%27s_constant`, Accessed: 2024-06-10.

Donald E. Knuth, *Analysis of the Euclidean Algorithm*, Journal of Computational Mathematics, vol. 12, no. 3, pp. 189–196, 1976. Available at: `https://pdf.sciencedirectassets.com/271503/1-s2.0-S0898122100X02179/1-s2.0-0898122176900250/main.pdf`