

Assignment 1

Shridhar Sharma — IMT2020065
Mayank Chadha — IMT2020045

Instructor: Prof. G.Srinivasaraghavan

Date: March 31, 2022

1 Description

Generating digits of π up to any given precision (number of decimal digits after the decimal point) using [Borwein's Algorithm](#).

*We could get **1000** correct digits in **9** iterations with the code running for **2.08 seconds** on our laptop.*

NOTE: Our code works correctly only upto Base 2^{13} . For Base $\geq 2^{14}$ integer overflow occurs, which is a hardware limitation.

For the Internal sub-parts of the assignment, we are comparing [Correct Digits of \$\pi\$](#) and [Correct Digits of \$\sqrt{2}\$](#) which are stored in tester.hpp to test them with our final output to ease the process of assignment checking.

We are using C++ because of its unique STL features such as [Vector and its properties](#).

Algorithm used for:

Addition - Addition Algorithm as taught in class.

Subtraction - Subtraction Algorithm as taught in class.

Multiplication - Both Karatsuba's Algorithm and Simple MUL algorithm as taught in class

Division - Division Algorithm as taught in class.

Square Root - Newton-Raphson Iterative method.

2 I/O and Commands

Inputs:

A Menu-driven interface is provided.

The internal Base used can be changed for rigorous code testing by modifying the "BASE" and "BASE_BITS" macros defined in pi.hpp. We are currently using Base 2^6 for best performance.

Outputs:

The digits of π or $\sqrt{2}$ are displayed on the screen with the precision and the correct number of digits. Also, the execution time is displayed.

Execution time varies according to the System Specifications.

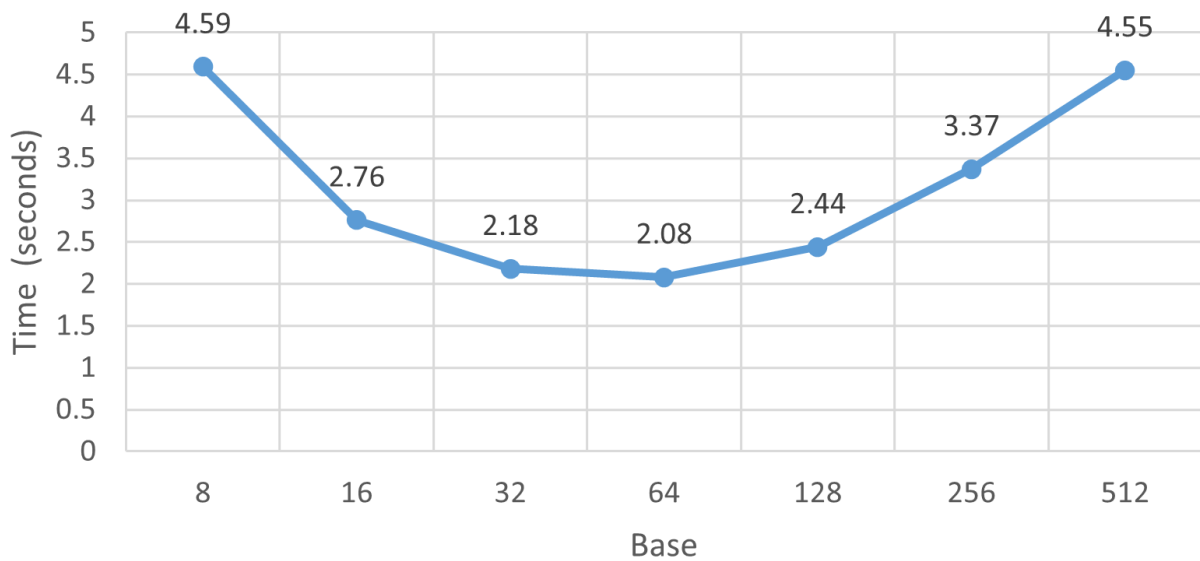
Commands to Compile and Run our Code:

```
1 make
2 ./pi
```

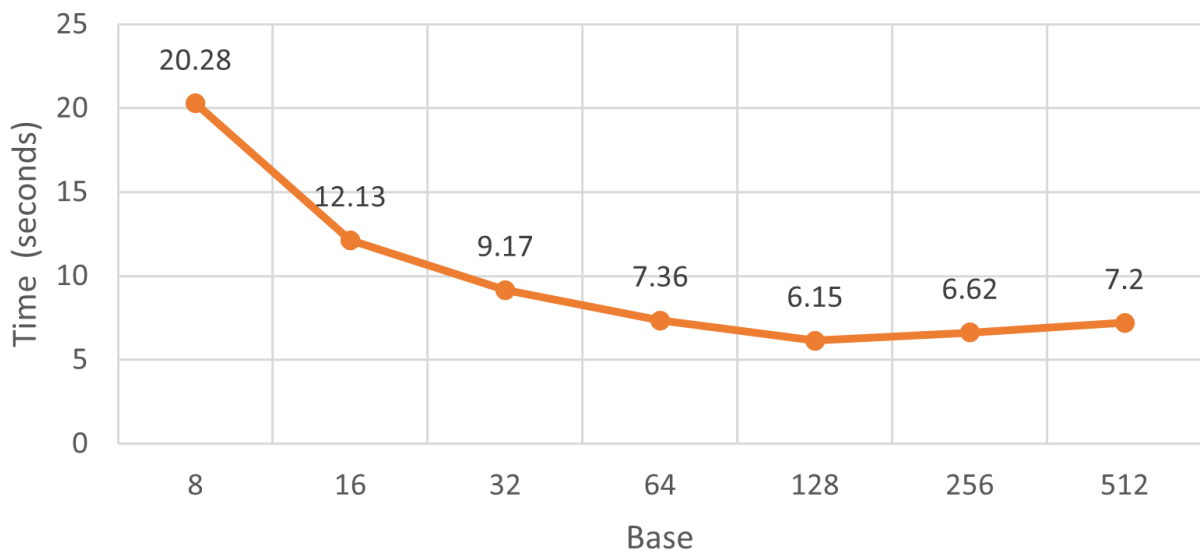
3 Observations

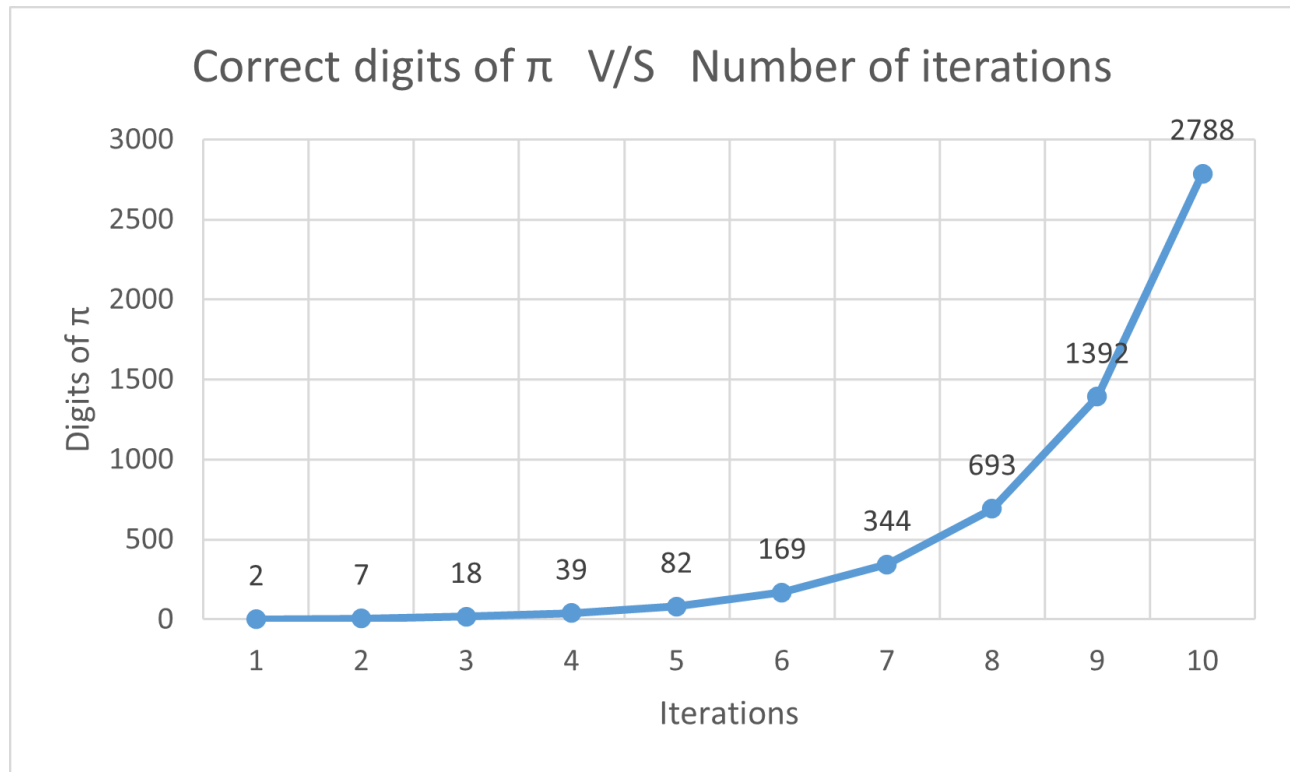
Some Graphs to help you interpret how some of the critical parameters vary with the execution of our code.

Time taken to calculate 1000 digits of π (using normal MUL)



Time taken to calculate 1000 digits of π (using Karatsuba)





```

2/2 + TiliX: gandalf@gandalf-omen:run/media/gandalf/New Volume/code/number_theory
> make
g++ main.cpp add.cpp sub.cpp mul.cpp div.cpp karatsuba.cpp helper.cpp sq_root.cpp pi.cpp -w -Wall -o pi
> ./pi
+-----+
| SM404 - NT : Assignment - 1 |
|                               |
| Shridhar Sharma (IMT2020065)|
| Mayank Chadha (IMT2020045) |
|                               |
| 1. SQ_ROOT(2)                |
| 2. PI (Karatsuba)            |
| 3. PI (Using normal MUL)     |
+-----+

Enter your choice: 3
Enter precision required (in decimal): 1000

3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808651328230664709384
4609550582231725359408128481117450284102701938521105559644622948954930381964428810975665933446128475648233786783165271201909145
6485669234603486104543266482133936072602491412737245870066063155881748815209209628292540917153643678925903600113305305488204665
2138414695194151160943305727036575959195309218611738193261179310511854807446237996274956735188575272489122793818301194912983367
3362440656643086021394946395224737190702179860943702770539217176293176752384674818467669405132000568127145263560827785771342757
7896091736371787214684409012249534301465495853710507922796892589235420199561121290219608640344181598136297747713099605187072113
4999999837297804995105973173281609631859502445945534690830264252230825334468503526193118817101000313783875288658753320838142061
717766914730359825349042875546873115956286388235378759375195778185778053217122680661300192787661119590921642019893809525

+-----+
Time taken to calculate pi: 3.70008 seconds.
Time taken to convert pi to decimal: 0.394329 seconds.

Iterations: 10    Current Precision: 1007
+-----+
Δ > /run/media/gandalf/New Volume/code/number_theory 10s 07:20:49 PM

```

Figure 1: Sample I/O