Cryptography 1/15

Reagan Shirk

January 15, 2020

Easy vs Hard

- We want computation for the user to be easy, hard for other people
- We want a large gap between easiness and hardness
  - Easy is a few seconds and hard is a million years
- How do we measure this and make it precise?

	Easy	Hard
Asymptotic complexity	P	Exponential time
Bit complexity	2^{40}	2^{80}
Time	1 year = 2^{25} seconds	1.8×10^{10} years = 2^{59} seconds

- If you want something that is considered safe, it needs to be at least 2^{80} (bit complexity), kind of standard

A Simple Problem: Addition and Multiplication

- Instinct says that multiplication should be *a little* harder than addition, but it's not actuallly? I couldn't tell what he said
- Integer representation: how to you put a given integer into a computer?
  - Size of the representation is important because it's how you determine asymptotic complexity
  - In everyday life, we use base 10 to describe numbers
    - For exapmle, our class number is 4823 (four thousand eight hundred twenty three), but it's of size 4
  - Base 2: You have integer N = $2^{log_2(N)}$, the size of the representation is $log_2(N)$

```
for i in range(2, N - 1)
if N % 1 == 0
    print(i)
```

  - What is the time complexity of the above code chunk? $O(Nlog^2N) \rightarrow$ exponential time because the size of the input is *log*
    - very bad algorithm
- Something about factoring...
  - If a number is not prime, then it must have a factor $< \sqrt{n}$
  - So maaayyyybe we could change the above algorithm to:

```
for i in range(2, ceil(sqrt(N)))
if N % 1 == 0
    print(i)
```

  - which would possibly change it to polynomial time
  - Now it takes $O(\sqrt{N}log^2(N))$ because $\sqrt{N} = 2^{\frac{log_2(N)}{2}}$
    * Still not good enough though

- Finally time for addition!
  - You have a number n with however many bits, your bit complexity is $O(n) \rightarrow$ easy and good

- Multiplication

```
def multi(M, N):
mul = 0
for i in range(N):
    mul = mul + M
print(mul)
```

  - Is this an algorithm?
    * Termination, check
    * Correctness, check
    * Complexity: $O(N(log(N) + log(M)) \rightarrow$ bit complexity, very bad. Exponential