CODING TEST 2019

Cryptography Lab

Task

Input:

- 16 number of N-bit positive integers (N >= 1000) A_1, ... , A_16

Output:

- B = A_1 + … + A_16

Allowed Instructions

- AVX2
- Multithreading
- Assembly

Big Integers

You have to design Big Integer class by yourself

AVX2

```
uint64_t *a, *b, *c;

__m256 a4 = _mm256_loadu_si256( (__m256i*)a );

__m256 b4 = _mm256_loadu_si256( (__m256i*)b );

__m256 c4 = _mm256_add_epi64( a4, b4 );

_mm256_storeu_si256( (__m256i*)c, c4 );
```

- compile with -mavx flag

Multithreading

```
#pragma omp parallel for num_threads(16)

for (int i = 0; i < n; i++) {

        do_smth();

}
```

- compile with -fopenmp and -lpthread flags

Evaluation Criteria

- Correctness (of course)
- w/o NTL library
- Speed

Schedule

- Deadline = **Feb. 22, 23:59 p.m.**
- Intermediate Meeting
  - Feb. 19 13:00 p.m. ~ 17:00 p.m.
  - Application due ~ Feb. 18 18:00 p.m. via email
-

Contact

- Andrey Kim - kimandrik@snu.ac.kr
- Duheong Kim - doodoo1204@snu.ac.kr
- Seungwan Hong - swanhong@snu.ac.kr

Optional Task

Multiplication of Big Integer by uint64_t

Summary of Tasks

A_i = big integer (more than 1000 bits)

a = small integer (less than 64 bits)

1. (Main task) Compute A_1 + … + A_16
2. (Optional task) Compute A x a