1 Git

5. Run "./homework-1 8" to check out the answer!
6. Run "./homework-1 8" to check out the answer!
7. Run "./homework-1 8" to check out the answer!

2 Floating-point arithmetics

1. An IEEE 64-bit floating-point value has 53 bits of mantissa (1 bit is implied), so the largest integer that can be represented as a 64-bit floating point WITHOUT GAP is 2^{53}. $2^{53}+1$ cannot be stored, because it exceeds the 52-bits mantissa allowance. Precisely, the spacing between 2^{52} and 2^{53} is 1 (i.e. every integers in this range can be represented under the IEEE 64-bit system), while the spacing between 2^{53} and 2^{54} is 2 (i.e. every even integers between this range can be represented). Run "./homework-1 1" to check out the answer!

   Alternatively, the largest integer that can be represented is DBL_MAX in C++. The largest possible exponent is 1023, and the largest mantissa is $1 + \sum_{i=1}^{52} 2^{-i} = 2 - 2^{-52}$, so the largest possible integer is DBL_MAX $= (2 - 2^{-52})2^{1023} = (2^{53} - 1) \times 2^{971}$.

2. An example of subnormal number is 2^{-1023}. Alternatively, run "./homework-1 2" to check out the answer!

3. Denote z to be the number you want to check. Run "./homework-1 3 z" to check out the answer!

3 Cache efficiency

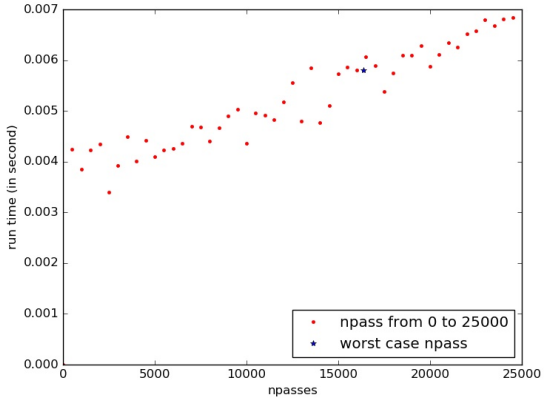
1. For a 12-way associative, 6MiB = 6291456 bytes L3 cache with a 64-byte cache line, there are $6291456/64 = 98304$ cache lines, and $98304/12 = 8192$ cache blocks. Given that an int takes up 4 bytes (in C++), each cache line can store $64/4 = 16$ integers, and each cache block has $16 \times 12 = 192$ integers.

   The worst-case *npasses* occurs when we take in each cache line, and have space in some other blocks. So the worst-case is $npasses = 16 \times 8192n + l = 131072n + l$ where $n \in \mathbb{N}$ and $l \in \{0, 1\}$.

2. On my machine, it has a 16-way associative, 8192K bytes L3 cache with 64-byte cache line. There are $8192 \times 2^{10}/64 = 131072$ cache lines, and $131072/16 = 8192$ cache blocks. Given that an int takes up 24 bytes (in Python 2.7), each cache line can store $\lfloor 64/24 \rfloor = 2$ integers.

   The worst-case *npasses* occurs when we take in each cache line, and have space in some other blocks. So the worst-case is $npasses = 2 \times 8192n + l = 16384n + l$ where $n \in \mathbb{N}$ and $l \in \{0, 1\}$.

3. The array has a size of $100,000$.


0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
run time (in second)
0
5000
10000
15000
20000
25000
npasses
npass from 0 to 25000
worst case npass