

CS69001 Computing Laboratory – I

Assignment No: B1

Date: 06–August–2018

This exercise deals with the printing of a set of proper fractions $a/b \in [0, 1)$ with $\gcd(a, b) = 1$. The user supplies a bound N on the denominator. We restrict only to fractions a/b satisfying $0 \leq a < b \leq N$. The printing should be in the ascending order of the (floating-point) values of a/b . Only the fractions in lowest terms are to be printed (each only once). For example, $1/3$ should be printed, but not $2/6, 3/9, 4/12, \dots$

We can populate an array A with these fractions, and sort the array to solve the problem. There are about $3N^2/\pi^2$ fractions under the given constraints, that is, the array A requires $\Theta(N^2)$ space. If you use an optimal sorting algorithm, the running time is $O(N^2 \log N)$. In this assignment, you are required to solve this problem in the same time, but using only $\Theta(N)$ additional space.

The algorithm uses a **min**-priority queue Q instead of the array A . The maximum size of Q will be N , so the space requirement is met. Moreover, insertion and deletion in Q can be designed so as to run in $O(\log N)$ time each. Since $\Theta(N^2)$ fractions are inserted and deleted, $\Theta(N^2 \log N)$ running time is achieved.

Part 1: The queue Q stores triples (a, b, f) , where a, b are integers in the range $0 \leq a < b \leq N$, and f is the floating-point value of the fraction a/b . Heap ordering is with respect to the f components of the triples. Write the min-priority-queue functions **heapify**, **buildheap**, **findmin**, **insert**, and **deletemin** for Q .

Part 2: The algorithm is based upon the insight that a/b and a'/b with $a \neq a'$ cannot be both smallest at the same time. Therefore Q will store (at most) one fraction of the form a/b for any given b .

Populate Q (as an array) with $0/1$, and with the $N - 1$ fractions $1/b$ for $b = 2, 3, \dots, N$. Convert Q to a heap.

Subsequently, run a loop so long as Q is not empty. Let the minimum stored in Q be (a, b, f) . Print it. Make a **deletemin** on Q . Try to locate the smallest integer a' in the range $a < a' < N$ with $\gcd(a', b) = 1$. If no such a' is found, we are done with this b . If a' is found, insert (a', b, f') in Q , where f' is the floating-point value of a'/b .

Write a function **printfractions** implementing this algorithm.

The *main()* function

- Read N from the user.
- Call **printfractions**.

Submit a single C/C++ source file. Do not use global/static variables. Do not invoke STL features.

Sample output

```
N = 16

+++ Fraction No  1 :    0 /  1 = 0.0000000000000000
+++ Fraction No  2 :    1 / 16 = 0.0625000000000000
+++ Fraction No  3 :    1 / 15 = 0.0666666666666667
+++ Fraction No  4 :    1 / 14 = 0.071428571428571
+++ Fraction No  5 :    1 / 13 = 0.076923076923077
+++ Fraction No  6 :    1 / 12 = 0.0833333333333333
+++ Fraction No  7 :    1 / 11 = 0.090909090909091
+++ Fraction No  8 :    1 / 10 = 0.1000000000000000
+++ Fraction No  9 :    1 /  9 = 0.1111111111111111
+++ Fraction No 10 :    1 /  8 = 0.1250000000000000
+++ Fraction No 11 :    2 / 15 = 0.1333333333333333
+++ Fraction No 12 :    1 /  7 = 0.142857142857143
+++ Fraction No 13 :    2 / 13 = 0.153846153846154
+++ Fraction No 14 :    1 /  6 = 0.1666666666666667
```

```

+++ Fraction No 15 : 2 / 11 = 0.181818181818182
+++ Fraction No 16 : 3 / 16 = 0.187500000000000
+++ Fraction No 17 : 1 / 5 = 0.200000000000000
+++ Fraction No 18 : 3 / 14 = 0.214285714285714
+++ Fraction No 19 : 2 / 9 = 0.222222222222222
+++ Fraction No 20 : 3 / 13 = 0.230769230769231
+++ Fraction No 21 : 1 / 4 = 0.250000000000000
+++ Fraction No 22 : 4 / 15 = 0.266666666666667
+++ Fraction No 23 : 3 / 11 = 0.272727272727273
+++ Fraction No 24 : 2 / 7 = 0.285714285714286
+++ Fraction No 25 : 3 / 10 = 0.300000000000000
+++ Fraction No 26 : 4 / 13 = 0.307692307692308
+++ Fraction No 27 : 5 / 16 = 0.312500000000000
+++ Fraction No 28 : 1 / 3 = 0.333333333333333
+++ Fraction No 29 : 5 / 14 = 0.357142857142857
+++ Fraction No 30 : 4 / 11 = 0.363636363636364
+++ Fraction No 31 : 3 / 8 = 0.375000000000000
+++ Fraction No 32 : 5 / 13 = 0.384615384615385
+++ Fraction No 33 : 2 / 5 = 0.400000000000000
+++ Fraction No 34 : 5 / 12 = 0.416666666666667
+++ Fraction No 35 : 3 / 7 = 0.428571428571429
+++ Fraction No 36 : 7 / 16 = 0.437500000000000
+++ Fraction No 37 : 4 / 9 = 0.444444444444444
+++ Fraction No 38 : 5 / 11 = 0.454545454545455
+++ Fraction No 39 : 6 / 13 = 0.461538461538462
+++ Fraction No 40 : 7 / 15 = 0.466666666666667
+++ Fraction No 41 : 1 / 2 = 0.500000000000000
+++ Fraction No 42 : 8 / 15 = 0.533333333333333
+++ Fraction No 43 : 7 / 13 = 0.538461538461538
+++ Fraction No 44 : 6 / 11 = 0.545454545454545
+++ Fraction No 45 : 5 / 9 = 0.555555555555556
+++ Fraction No 46 : 9 / 16 = 0.562500000000000
+++ Fraction No 47 : 4 / 7 = 0.571428571428571
+++ Fraction No 48 : 7 / 12 = 0.583333333333333
+++ Fraction No 49 : 3 / 5 = 0.600000000000000
+++ Fraction No 50 : 8 / 13 = 0.615384615384615
+++ Fraction No 51 : 5 / 8 = 0.625000000000000
+++ Fraction No 52 : 7 / 11 = 0.636363636363636
+++ Fraction No 53 : 9 / 14 = 0.642857142857143
+++ Fraction No 54 : 2 / 3 = 0.666666666666667
+++ Fraction No 55 : 11 / 16 = 0.687500000000000
+++ Fraction No 56 : 9 / 13 = 0.692307692307692
+++ Fraction No 57 : 7 / 10 = 0.700000000000000
+++ Fraction No 58 : 5 / 7 = 0.714285714285714
+++ Fraction No 59 : 8 / 11 = 0.727272727272727
+++ Fraction No 60 : 11 / 15 = 0.733333333333333
+++ Fraction No 61 : 3 / 4 = 0.750000000000000
+++ Fraction No 62 : 10 / 13 = 0.769230769230769
+++ Fraction No 63 : 7 / 9 = 0.777777777777778
+++ Fraction No 64 : 11 / 14 = 0.785714285714286
+++ Fraction No 65 : 4 / 5 = 0.800000000000000
+++ Fraction No 66 : 13 / 16 = 0.812500000000000
+++ Fraction No 67 : 9 / 11 = 0.818181818181818
+++ Fraction No 68 : 5 / 6 = 0.833333333333333
+++ Fraction No 69 : 11 / 13 = 0.846153846153846
+++ Fraction No 70 : 6 / 7 = 0.857142857142857
+++ Fraction No 71 : 13 / 15 = 0.866666666666667
+++ Fraction No 72 : 7 / 8 = 0.875000000000000
+++ Fraction No 73 : 8 / 9 = 0.888888888888889
+++ Fraction No 74 : 9 / 10 = 0.900000000000000
+++ Fraction No 75 : 10 / 11 = 0.909090909090909
+++ Fraction No 76 : 11 / 12 = 0.916666666666667
+++ Fraction No 77 : 12 / 13 = 0.923076923076923
+++ Fraction No 78 : 13 / 14 = 0.928571428571429
+++ Fraction No 79 : 14 / 15 = 0.933333333333333
+++ Fraction No 80 : 15 / 16 = 0.937500000000000

```