

Name: Jude Onyia  
 Student ID: V00947095  
 Email: judeonyia10@gmail.com  
 Course: ECE596C  
 Section: T01

Assignment ID: cpp\_arithmetic  
 Assignment Title: Interval Arithmetic and Applications

Submission Source: [https://github.com/uvic-seng475-2020-05/cpp\\_arithmetic-JudeOnyia.git](https://github.com/uvic-seng475-2020-05/cpp_arithmetic-JudeOnyia.git)

Commit ID: 301c29dfd89ae92f8524e4fb633cefe4c7df11b1

#### Submitted Files

=====

```
drwxrwxr-x      135 2020-06-15 00:50 ./app
-rw-rw-r--     4120 2020-06-15 00:50 ./app/delaunay_triangulation.cpp
-rw-rw-r--     2427 2020-06-15 00:50 ./app/test_interval.cpp
-rw-rw-r--     3586 2020-06-15 00:50 ./app/test_kernel.cpp
-rw-rw-r--    22880 2020-06-15 00:50 ./app/triangulation_2.hpp
-rw-rw-r--      740 2020-06-15 00:50 ./CMakeLists.txt
-rw-rw-r--     144 2020-06-15 00:50 ./IDENTIFICATION.txt
drwxrwxr-x      24 2020-06-15 00:50 ./include
drwxrwxr-x      56 2020-06-15 00:50 ./include/ra
-rw-rw-r--     6290 2020-06-15 00:50 ./include/ra/interval.hpp
-rw-rw-r--    11224 2020-06-15 00:50 ./include/ra/kernel.hpp
-rw-rw-r--   113754 2020-06-15 00:50 ./README.pdf
```

#### Results

=====

Package	Operation	Target	Status
nonprog	generate	---	OK (0.0s)
interval_orig	generate	---	OK (0.3s)
interval_orig	configure	---	OK (2.3s)
interval_orig	build	test_interval	OK (0.8s)
interval_orig	build	test_kernel	OK (5.2s)
interval_sane	generate	---	OK (0.4s)
interval_sane	configure	---	OK (2.1s)
interval_sane	build	test_interval	FAIL (2 0.9s 95L)
interval_sane	build	test_kernel	FAIL (2 4.5s 183L)
deltri_orig	generate	---	OK (0.4s)
deltri_orig	configure	---	OK (2.1s)
deltri_orig	build	delaunay_triangulation	OK (8.2s)

Normally, an operation is indicated as having a status of either "OK" or "FAIL". A status of "?" indicates that the operation could not be performed for some reason (e.g., due to an earlier error or being a manual step). The time (in seconds) required for an operation is denoted by an expression consisting of a number followed by the letter "s" (e.g., "5.0s"). In the case of a test that consists of multiple test cases, the number of failed test cases and total number of test cases is expressed as a fraction (e.g., "10/50" means 10 test cases failed out of 50 test cases in total). The length (in lines) of the log file generated by an operation is denoted by an expression consisting of a number followed by the letter "L" (e.g., "10L"). To ascertain the reason for the failure of an operation, check the contents of the log file provided.

## Legend

=====

Package: nonprog

Nonprogramming exercises

Package: interval\_orig

The code as originally submitted by the student.

Build target: test\_interval

Build the test\_interval program.

Build target: test\_kernel

Build the test\_kernel program.

Package: interval\_sane

Code with modifications to perform API sanity checking.

Build target: test\_interval

Build the (dummy) test\_interval program.

Build target: test\_kernel

Build the (dummy) test\_kernel program.

Package: deltri\_orig

The code as originally submitted by the student.

Build target: delaunay\_triangulation

Build the delaunay\_triangulation program.

```
1 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake
2 -S/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J
3 udeOnyia/cktmp/package-interval_sane/source
4 -B/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J
5 udeOnyia/cktmp/package-interval_sane/derived --check-build-system
6 CMakeFiles/Makefile.cmake 0
7 /usr/bin/gmake -f CMakeFiles/Makefile2 test_interval
8 gmake[1]: Entering directory
9 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
10 deOnyia/cktmp/package-interval_sane/derived'
11 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake
12 -S/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J
13 udeOnyia/cktmp/package-interval_sane/source
14 -B/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J
15 udeOnyia/cktmp/package-interval_sane/derived --check-build-system
16 CMakeFiles/Makefile.cmake 0
17 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -E
18 cmake_progress_start
19 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
20 eOnyia/cktmp/package-interval_sane/derived/CMakeFiles 2
21 /usr/bin/gmake -f CMakeFiles/Makefile2 CMakeFiles/test_interval.dir/all
22 gmake[2]: Entering directory
23 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
24 deOnyia/cktmp/package-interval_sane/derived'
25 /usr/bin/gmake -f CMakeFiles/test_interval.dir/build.make
26 CMakeFiles/test_interval.dir/depend
27 gmake[3]: Entering directory
28 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
29 deOnyia/cktmp/package-interval_sane/derived'
30 cd
31 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
32 eOnyia/cktmp/package-interval_sane/derived &&
33 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -E
34 cmake_depends "Unix Makefiles"
35 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
36 eOnyia/cktmp/package-interval_sane/source
37 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
38 eOnyia/cktmp/package-interval_sane/source
39 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
40 eOnyia/cktmp/package-interval_sane/derived
41 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
42 eOnyia/cktmp/package-interval_sane/derived
43 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
44 eOnyia/cktmp/package-interval_sane/derived/CMakeFiles/test_interval.dir/DependIn
45 fo.cmake --color=
46 Scanning dependencies of target test_interval
47 gmake[3]: Leaving directory
48 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
49 deOnyia/cktmp/package-interval_sane/derived'
50 /usr/bin/gmake -f CMakeFiles/test_interval.dir/build.make
51 CMakeFiles/test_interval.dir/build
52 gmake[3]: Entering directory
53 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
54 deOnyia/cktmp/package-interval_sane/derived'
55 [ 50%] Building CXX object CMakeFiles/test_interval.dir/app/test_interval.cpp.o
56 /home/frodo/public/ugls_lab-4.0.70/bin/c++
57 -I/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J
58 udeOnyia/cktmp/package-interval_sane/source/include -pedantic-errors
59 -frounding-math -std=gnu++17 -o
60 CMakeFiles/test_interval.dir/app/test_interval.cpp.o -c
61 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
62 eOnyia/cktmp/package-interval_sane/source/app/test_interval.cpp
```

```
63 In file included from
64 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
65 eOnyia/cktmp/package-interval_sane/source/app/test_interval.cpp:4:
66 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
67 eOnyia/cktmp/package-interval_sane/source/include/ra/interval.hpp:116:73:
68 error: C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@
M-^Y or
69 âM-^@M-^X-std=gnu++2aâM-^@M-^Y [-Wpedantic]
70     116 |     typename interval<real_type>::statistics interval<real_type>::stat_ =
71     {.indeterminate_result_count = 0, .arithmetic_op_count = 0};
72     |
73     ^
74 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
75 eOnyia/cktmp/package-interval_sane/source/include/ra/interval.hpp:116:106:
76 error: C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@
M-^Y or
77 âM-^@M-^X-std=gnu++2aâM-^@M-^Y [-Wpedantic]
78     116 |     typename interval<real_type>::statistics interval<real_type>::stat_ =
79     {.indeterminate_result_count = 0, .arithmetic_op_count = 0};
80     |
81     ^
82 gmake[3]: *** [CMakeFiles/test_interval.dir/app/test_interval.cpp.o] Error 1
83 gmake[3]: Leaving directory
84 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
85 deOnyia/cktmp/package-interval_sane/derived'
86 gmake[2]: *** [CMakeFiles/test_interval.dir/all] Error 2
87 gmake[2]: Leaving directory
88 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
89 deOnyia/cktmp/package-interval_sane/derived'
90 gmake[1]: *** [CMakeFiles/test_interval.dir/rule] Error 2
91 gmake[1]: Leaving directory
92 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
93 deOnyia/cktmp/package-interval_sane/derived'
94 gmake: *** [test_interval] Error 2
95 ERROR: build failed to generate executable test_interval
```

```
1 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake
2 -S/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J
3 udeOnyia/cktmp/package-interval_sane/source
4 -B/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J
5 udeOnyia/cktmp/package-interval_sane/derived --check-build-system
6 CMakeFiles/Makefile.cmake 0
7 /usr/bin/gmake -f CMakeFiles/Makefile2 test_kernel
8 gmake[1]: Entering directory
9 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
10 deOnyia/cktmp/package-interval_sane/derived'
11 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake
12 -S/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J
13 udeOnyia/cktmp/package-interval_sane/source
14 -B/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J
15 udeOnyia/cktmp/package-interval_sane/derived --check-build-system
16 CMakeFiles/Makefile.cmake 0
17 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -E
18 cmake_progress_start
19 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
20 eOnyia/cktmp/package-interval_sane/derived/CMakeFiles 2
21 /usr/bin/gmake -f CMakeFiles/Makefile2 CMakeFiles/test_kernel.dir/all
22 gmake[2]: Entering directory
23 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
24 deOnyia/cktmp/package-interval_sane/derived'
25 /usr/bin/gmake -f CMakeFiles/test_kernel.dir/build.make
26 CMakeFiles/test_kernel.dir/depend
27 gmake[3]: Entering directory
28 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
29 deOnyia/cktmp/package-interval_sane/derived'
30 cd
31 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
32 eOnyia/cktmp/package-interval_sane/derived &&
33 /home/frodo/public/ugls_lab-4.0.70/packages/cmake-3.17.1/bin/cmake -E
34 cmake_depends "Unix Makefiles"
35 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
36 eOnyia/cktmp/package-interval_sane/source
37 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
38 eOnyia/cktmp/package-interval_sane/source
39 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
40 eOnyia/cktmp/package-interval_sane/derived
41 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
42 eOnyia/cktmp/package-interval_sane/derived
43 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
44 eOnyia/cktmp/package-interval_sane/derived/CMakeFiles/test_kernel.dir/DependInfo
45 .cmake --color=
46 Scanning dependencies of target test_kernel
47 gmake[3]: Leaving directory
48 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
49 deOnyia/cktmp/package-interval_sane/derived'
50 /usr/bin/gmake -f CMakeFiles/test_kernel.dir/build.make
51 CMakeFiles/test_kernel.dir/build
52 gmake[3]: Entering directory
53 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
54 deOnyia/cktmp/package-interval_sane/derived'
55 [ 50%] Building CXX object CMakeFiles/test_kernel.dir/app/test_kernel.cpp.o
56 /home/frodo/public/ugls_lab-4.0.70/bin/c++
57 -I/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-J
58 udeOnyia/cktmp/package-interval_sane/source/include -pedantic-errors
59 -frounding-math -std=gnu++17 -o
60 CMakeFiles/test_kernel.dir/app/test_kernel.cpp.o -c
61 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
62 eOnyia/cktmp/package-interval_sane/source/app/test_kernel.cpp
```

```

63 In file included from
64 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
65 eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:6,
66 from
67 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
68 eOnyia/cktmp/package-interval_sane/source/app/test_kernel.cpp:3:
69 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
70 eOnyia/cktmp/package-interval_sane/source/include/ra/interval.hpp:116:73:
71 error: C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@
M-^Y or
72 âM-^@M-^X-std=gnu++2aâM-^@M-^Y [-Wpedantic]
73 116 | typename interval<real_type>::statistics interval<real_type>::stat_ =
74 {.indeterminate_result_count = 0, .arithmetic_op_count = 0};
75 |
76 ^
77 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
78 eOnyia/cktmp/package-interval_sane/source/include/ra/interval.hpp:116:106:
79 error: C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@
M-^Y or
80 âM-^@M-^X-std=gnu++2aâM-^@M-^Y [-Wpedantic]
81 116 | typename interval<real_type>::statistics interval<real_type>::stat_ =
82 {.indeterminate_result_count = 0, .arithmetic_op_count = 0};
83 |
84 ^
85 In file included from
86 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
87 eOnyia/cktmp/package-interval_sane/source/app/test_kernel.cpp:3:
88 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
89 eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:25:35: warning:
90 declaration âM-^@M-^Xstruct ra::math::indeterminate_resultâM-^@M-^Y does not dec
lare anything
91 25 | using idr = typename ra::math::indeterminate_result;
92 | ^~~~~~
93 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
94 eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:27:32: warning:
95 declaration âM-^@M-^Xclass CGAL::MP_FloatâM-^@M-^Y does not declare anything
96 27 | using exct = typename CGAL::MP_Float;
97 | ^~~~~~
98 In file included from
99 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
100 eOnyia/cktmp/package-interval_sane/source/app/test_kernel.cpp:3:
101 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
102 eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:253:59: error:
103 C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@M-^Y or
âM-^@M-^X-std=gnu++2aâM-^@M-^Y
104 [-Wpedantic]
105 253 | typename Kernel<Real>::Statistics Kernel<Real>::stat_ =
106 {.orientation_total_count = 0, .orientation_exact_count = 0,
107 .preferred_direction_total_count = 0, .preferred_direction_exact_count = 0,
108 .side_of_oriented_circle_total_count = 0, .side_of_oriented_circle_exact_count
109 = 0};
110 |
111 ^
112 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
113 eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:253:89: error:
C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@M-^Y or
âM-^@M-^X-std=gnu++2aâM-^@M-^Y
114 [-Wpedantic]
115 253 | typename Kernel<Real>::Statistics Kernel<Real>::stat_ =
116 {.orientation_total_count = 0, .orientation_exact_count = 0,
117 .preferred_direction_total_count = 0, .preferred_direction_exact_count = 0,
118 .side_of_oriented_circle_total_count = 0, .side_of_oriented_circle_exact_count
119 = 0};

```

```
120 |
121 ^
122 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
123 eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:253:119: error:
124 C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@M-^Y or
125 âM-^@M-^X-std=gnu++2aâM-^@M-^Y
126 [-Wpedantic]
127 253 | typename Kernel<Real>::Statistics Kernel<Real>::stat_ =
128     | { .orientation_total_count = 0, .orientation_exact_count = 0,
129     |   .preferred_direction_total_count = 0, .preferred_direction_exact_count = 0,
130     |   .side_of_oriented_circle_total_count = 0, .side_of_oriented_circle_exact_count
131     | = 0 };
132 |
133 ^
134 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
135 eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:253:157: error:
136 C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@M-^Y or
137 âM-^@M-^X-std=gnu++2aâM-^@M-^Y
138 [-Wpedantic]
139 253 | typename Kernel<Real>::Statistics Kernel<Real>::stat_ =
140     | { .orientation_total_count = 0, .orientation_exact_count = 0,
141     |   .preferred_direction_total_count = 0, .preferred_direction_exact_count = 0,
142     |   .side_of_oriented_circle_total_count = 0, .side_of_oriented_circle_exact_count
143     | = 0 };
144 |
145 ^
146 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
147 eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:253:195: error:
148 C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@M-^Y or
149 âM-^@M-^X-std=gnu++2aâM-^@M-^Y
150 [-Wpedantic]
151 253 | typename Kernel<Real>::Statistics Kernel<Real>::stat_ =
152     | { .orientation_total_count = 0, .orientation_exact_count = 0,
153     |   .preferred_direction_total_count = 0, .preferred_direction_exact_count = 0,
154     |   .side_of_oriented_circle_total_count = 0, .side_of_oriented_circle_exact_count
155     | = 0 };
156 |
157 ^
158 /home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Jud
159 eOnyia/cktmp/package-interval_sane/source/include/ra/kernel.hpp:253:237: error:
160 C++ designated initializers only available with âM-^@M-^X-std=c++2aâM-^@M-^Y or
161 âM-^@M-^X-std=gnu++2aâM-^@M-^Y
162 [-Wpedantic]
163 253 | typename Kernel<Real>::Statistics Kernel<Real>::stat_ =
164     | { .orientation_total_count = 0, .orientation_exact_count = 0,
165     |   .preferred_direction_total_count = 0, .preferred_direction_exact_count = 0,
166     |   .side_of_oriented_circle_total_count = 0, .side_of_oriented_circle_exact_count
167     | = 0 };
168 |
169 ^
170 gmake[3]: *** [CMakeFiles/test_kernel.dir/app/test_kernel.cpp.o] Error 1
171 gmake[3]: Leaving directory
172 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
173 deOnyia/cktmp/package-interval_sane/derived'
174 gmake[2]: *** [CMakeFiles/test_kernel.dir/all] Error 2
175 gmake[2]: Leaving directory
176 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
177 deOnyia/cktmp/package-interval_sane/derived'
```

```
178 gmake[1]: *** [CMakeFiles/test_kernel.dir/rule] Error 2
179 gmake[1]: Leaving directory
180 `/home/judeonyia/Documents/ECE596C_Assignments/ECE596C_Assgn_3/cpp_arithmetic-Ju
181 deOnyia/cktmp/package-interval_sane/derived'
182 gmake: *** [test_kernel] Error 2
183 ERROR: build failed to generate executable test_kernel
```



```
1  commit 2e5f4130237556ce16fe8499c8eb2cd218206a6c
2  Author: JudeOnyia <60678029+JudeOnyia@users.noreply.github.com>
3  Date:   Thu Jun 11 01:23:01 2020 -0700
4
5      First Commit
6
7  commit 0fd81b4d2225985328adcc597aaf191bb0cb2421
8  Author: Jude Onyia <judeonyia10@gmail.com>
9  Date:   Thu Jun 11 18:12:27 2020 -0700
10
11     1) wrote indeterminate_result class
12     2) wrote rounding_mode_saver class
13     3) wrote statistics class
14     4) wrote default constructor and compound operator overloads (+,-,*)
15     5) wrote two argument constructor
16     6) wrote upper, lower, is_singleton and sign member functions
17
18  commit 4d3078e339c39e99c68a742c23c559fa9c688e26
19  Author: Jude Onyia <judeonyia10@gmail.com>
20  Date:   Fri Jun 12 02:43:44 2020 -0700
21
22     1) Wrote clear_statistics member function
23     2) Wrote get_statistics member function
24     3) Wrote binary add, sub, mult non-member functions
25     4) Wrote less than operator overload
26     5) Wrote stream inserter
27
28  commit e75ac306de5ff98af6a01dcae95bf65a26f5400a
29  Author: JudeOnyia <60678029+JudeOnyia@users.noreply.github.com>
30  Date:   Fri Jun 12 15:34:29 2020 -0700
31
32     Starting Kernel class
33
34  commit 8d2796f420bc33bacd8221ec3cf952fec7cad7d0
35  Author: Jude Onyia <judeonyia10@gmail.com>
36  Date:   Fri Jun 12 20:58:07 2020 -0700
37
38     1) Wrote the types needed for the Kernel class
39     2) Wrote the Orientation and Oriented_side enum classes
40     3) Wrote Statistics struct
41     4) Wrote a template function to compute the determinant of a
42         3 by 3 matrix
43     5) Wrote the orientation member function that uses interval
44         arithmetic or exact arithmetic if interval yields indetermine
45         result
46
47  commit 68ffe60f61a40f2c717223f08d1813940de3ed7c
48  Author: Jude Onyia <judeonyia10@gmail.com>
49  Date:   Sat Jun 13 01:09:36 2020 -0700
50
51     1) Wrote static function for solving determinant of 4 by 4 matrix
52     2) Wrote side_of_oriented_circle member function
53
54  commit 4b77c1fe22e0c876a07c3a18482872e696002f97
55  Author: Jude Onyia <judeonyia10@gmail.com>
56  Date:   Sat Jun 13 18:55:44 2020 -0700
57
58     1) Wrote preferred_direction member function
59     2) Wrote is_strictly_convex_quad member function
60
61  commit 26168d5d9d4049b1fb342bc49a4ece6f58d15306
62  Author: Jude Onyia <judeonyia10@gmail.com>
```

```
63 Date: Sat Jun 13 23:11:03 2020 -0700
64
65 1) Wrote is_locally_delaunay_edge member function
66 2) Wrote is_locally_pd_delaunay_edge member function
67 3) Wrote clear_statistics member function
68 4) Wrote get_statistics member function
69
70 commit 5db9ce8ba4110ace1ff6126d5ceda2164c813dee
71 Author: Jude Onyia <judeonyia10@gmail.com>
72 Date: Sun Jun 14 00:52:55 2020 -0700
73
74 Added some static specifiers and did some const correctness
75
76 commit 87a35d587bab40497e037dcb34c70dbc2a67bb24
77 Author: JudeOnyia <60678029+JudeOnyia@users.noreply.github.com>
78 Date: Sun Jun 14 14:57:03 2020 -0700
79
80 Starting delaunay triangulation
81
82 commit f479bf137e763c2e18e916d5c3f47ece25eecd5
83 Author: Jude Onyia <judeonyia10@gmail.com>
84 Date: Sun Jun 14 22:20:15 2020 -0700
85
86 1) Wrote function to check if half edge is flippable
87 2) Modified is_strictly_convex function in kernel.hpp
88
89 commit 3813050045c41cffeade0ff47c11d88fc6816a55
90 Author: Jude Onyia <judeonyia10@gmail.com>
91 Date: Mon Jun 15 00:31:57 2020 -0700
92
93 1) applied the Lawson local optimization procedure
94
95 commit 301c29dfd89ae92f8524e4fb633cefe4c7df11b1
96 Author: JudeOnyia <60678029+JudeOnyia@users.noreply.github.com>
97 Date: Mon Jun 15 00:45:17 2020 -0700
98
99 1) Removed traingulation_2_demo
100 2) Commented out cmakelists line of demo
```

Name: Jude Onyia  
Student ID: V00947095  
Course: ECE 596C  
Due Date: June 19, 2020

### Assignment 3: Non – Programming Exercise

6.1)

Due to exceptions that could be thrown for several reasons (i.e. division by zero, lack of memory, etc.), performing any necessary clean-up of an object must be put in a finalizer of that object. When exception is thrown, the program is violently removed from that code block, it does not execute any code further down the line from that exception. The only code guaranteed to execute is the code within the finalizer of that object. Therefore, any clean-up necessary must be done in the finalizer.

6.2 a)

During the stack unwinding process, objects are destroyed in the following sequence: die3, die, countdown, hello, i, bjarne, herb, dv, u, z.

6.2 b)

During the stack unwinding process, the only object destroyed is s.

6.3 a)

If there is insufficient memory to allocate for second buffer, an exception will be thrown, and the program will be violently ripped from the function without freeing the first buffer. To guarantee that both buffers are freed if an exception occurs, one can use a class for the buffers, where a char pointer is a data member. Overloads on operator= and operator[] can be used to assign the char pointer to a space in memory and access it, respectively. The finalizer of the class can be used to safely deallocate the memory if an exception is thrown. Both buf1 and buf2 will then be objects of this class. In the incident where buf1 is created and memory runs out while attempting to create buf2, as the exception is thrown, it is guaranteed that the finalizer of buf1 will be called to free up that space before leaving the function.

6.3 b)

If the formatting flags are changed and outputting the integer to the ostream causes an exception to be thrown, the program will be violently ripped from the function without running the code that restores the formatting flag. To prevent this, the line that changes the formatting flag and outputs the integer can be surrounded by a try clause. The body of the associated catch clause can restore the old formatting flags if an exception occurs.

6.3 c)

If the queue has a number of elements in it, and there is insufficient amount of memory left to push another element, q.push\_back(value) might throw an exception. If it does and the user of the class does not catch it, the program could terminate without freeing the memory allocated for the elements already in the queue. To prevent this, a finalizer for the class can be explicitly defined to free

the queue when called. This will ensure that when an exception is thrown, the memory allocated for elements in the queue are freed before the object is destroyed and the program terminated.

6.5)

The function 'analyze' cannot throw an exception because the function has the noexcept specifier. This specifier indicates to the compiler that code for exceptions is not needed for this function, therefore, the compiler will not generate code for exceptions for this function. This means the function is incapable of throwing an exception. If an attempt is made to throw an exception in this noexcept function, this will result in a fatal error.

The function 'doWork' does not have a noexcept specifier, therefore, it may or may not throw an exception. The compiler will generate code for exceptions for this function. Therefore, this function is capable of throwing an exception.

```
1  # Specify Minimum Required Version
2  cmake_minimum_required(VERSION 3.1 FATAL_ERROR)
3
4  # Specify Project and Language
5  project(cpp_arithmetic LANGUAGES CXX)
6
7  # Set CXX Flags with the -frounding-math option
8  set(CMAKE_CXX_FLAGS "-frounding-math")
9
10 # Find CGAL Library
11 find_package(CGAL REQUIRED)
12
13 # Set Include directories and libraries
14 include_directories(include ${CGAL_INCLUDE_DIRS})
15 link_libraries(${CGAL_LIBRARY} ${GMP_LIBRARIES})
16
17 # Add Executable Program
18 add_executable(test_interval app/test_interval.cpp)
19 add_executable(test_kernel app/test_kernel.cpp)
20 #add_executable(triangulation_2_demo app/triangulation_2_demo.cpp app/triangulation_2.hpp)
21 add_executable(delaunay_triangulation app/delaunay_triangulation.cpp app/triangulation_2.hpp)
```

```

1  #ifndef INTERVAL_HPP
2  #define INTERVAL_HPP
3  #include<cfenv>
4  #include<stdexcept>
5  #include<algorithm>
6  #include<iostream>
7  namespace ra::math {
8
9      struct indeterminate_result : public std::runtime_error{
10         using std::runtime_error::runtime_error;
11     };
12
13     class rounding_mode_saver {
14     public:
15         rounding_mode_saver() : current_round_mode(std::fegetround()) {}
16         ~rounding_mode_saver() {
17             std::fesetround(current_round_mode);
18         }
19         rounding_mode_saver(rounding_mode_saver&&) = delete;
20         rounding_mode_saver(const rounding_mode_saver&) = delete;
21         rounding_mode_saver& operator=(rounding_mode_saver&&) = delete;
22         rounding_mode_saver& operator=(const rounding_mode_saver&) = delete;
23     private:
24         int current_round_mode;
25     };
26
27     template<class T>
28     class interval{
29     public:
30         using real_type = T;
31
32         struct statistics{
33             // The total number of indeterminate results encountered.
34             unsigned long indeterminate_result_count;
35             // The total number of interval arithmetic operations.
36             unsigned long arithmetic_op_count;
37         };
38
39         interval(real_type set_value = real_type(0)) : lower_(set_value), up
per_(set_value) {}
40         interval(real_type lower, real_type upper) : lower_((lower<upper)? l
ower:upper), upper_((upper>lower)? upper:lower) {}
41         real_type lower() const { return lower_; }
42         real_type upper() const { return upper_; }
43
44         static void incr_indeterminate_result_count() { ++(stat_.indetermina
te_result_count); }
45         static void incr_arithmetic_op_count() { ++(stat_.arithmetic_op_coun
t); }
46
47         interval& operator+=(const interval& obj){
48             real_type upper_Temp(0);
49             real_type lower_Temp(0);
50             rounding_mode_saver rms; // Save the rounding mode that should b
e restored
51             std::fesetround(FE_DOWNWARD);
52             lower_Temp = lower_ + obj.lower();
53             std::fesetround(FE_UPWARD);
54             upper_Temp = upper_ + obj.upper();
55             lower_ = lower_Temp;
56             upper_ = upper_Temp;
57             incr_arithmetic_op_count();

```

```

58         return *this;
59     }
60     interval& operator--=(const interval& obj){
61         real_type upper_Temp(0);
62         real_type lower_Temp(0);
63         rounding_mode_saver rms; // Save the rounding mode that should be
e restored
64         std::fesetround(FE_DOWNWARD);
65         lower_Temp = lower_ - obj.upper();
66         std::fesetround(FE_UPWARD);
67         upper_Temp = upper_ - obj.lower();
68         lower_ = lower_Temp;
69         upper_ = upper_Temp;
70         incr_arithmetic_op_count();
71         return *this;
72     }
73     interval& operator*=(const interval& obj){
74         real_type upper_Temp(0);
75         real_type lower_Temp(0);
76         rounding_mode_saver rms; // Save the rounding mode that should be
e restored
77         std::fesetround(FE_DOWNWARD);
78         lower_Temp = std::min(std::min((lower_*obj.lower()), (lower_*obj.
upper()))), std::min((upper_*obj.lower()), (upper_*obj.upper())));
79         std::fesetround(FE_UPWARD);
80         upper_Temp = std::max(std::max((lower_*obj.lower()), (lower_*obj.
upper()))), std::max((upper_*obj.lower()), (upper_*obj.upper())));
81         lower_ = lower_Temp;
82         upper_ = upper_Temp;
83         incr_arithmetic_op_count();
84         return *this;
85     }
86
87     bool is_singleton() const { return (lower_==upper_)? true : false; }
88
89     int sign() const {
90         if((lower_ < real_type(0)) && (upper_ < real_type(0))){ return -
1; }
91         else if((lower_ > real_type(0)) && (upper_ > real_type(0))){ ret
urn 1; }
92         else if((lower_ == real_type(0)) && (upper_ == real_type(0))){ r
eturn 0; }
93         else{
94             incr_indeterminate_result_count();
95             throw indeterminate_result("Indeterminate sign of interval");
96         }
97     }
98
99     static void clear_statistics() {
100         stat_.indeterminate_result_count = 0;
101         stat_.arithmetic_op_count = 0;
102     }
103
104     static void get_statistics(statistics& stat) {
105         stat = stat_;
106     }
107
108     private:
109     real_type upper_;
110     real_type lower_;
111     static statistics stat_;
112

```

```

113     };
114
115     template<class real_type>
116     typename interval<real_type>::statistics interval<real_type>::stat_ = {.inde
117     terminate_result_count = 0, .arithmetic_op_count = 0};
118
119     template<class real_type>
120     interval<real_type> operator+(const interval<real_type>& obj_A, const interv
121     al<real_type>& obj_B){
122         real_type upper_Temp(0);
123         real_type lower_Temp(0);
124         rounding_mode_saver rms; // Save the rounding mode that should be restor
125     ed
126         std::fesetround(FE_DOWNWARD);
127         lower_Temp = obj_A.lower() + obj_B.lower();
128         std::fesetround(FE_UPWARD);
129         upper_Temp = obj_A.upper() + obj_B.upper();
130         interval<real_type> result(lower_Temp,upper_Temp);
131         result.incr_arithmetic_op_count();
132         return result;
133     }
134
135     template<class real_type>
136     interval<real_type> operator-(const interval<real_type>& obj_A, const interv
137     al<real_type>& obj_B){
138         real_type upper_Temp(0);
139         real_type lower_Temp(0);
140         rounding_mode_saver rms; // Save the rounding mode that should be restor
141     ed
142         std::fesetround(FE_DOWNWARD);
143         lower_Temp = obj_A.lower() - obj_B.upper();
144         std::fesetround(FE_UPWARD);
145         upper_Temp = obj_A.upper() - obj_B.lower();
146         interval<real_type> result(lower_Temp,upper_Temp);
147         result.incr_arithmetic_op_count();
148         return result;
149     }
150
151     template<class real_type>
152     interval<real_type> operator*(const interval<real_type>& obj_A, const interv
153     al<real_type>& obj_B){
154         real_type upper_Temp(0);
155         real_type lower_Temp(0);
156         rounding_mode_saver rms; // Save the rounding mode that should be restor
157     ed
158         std::fesetround(FE_DOWNWARD);
159         lower_Temp = std::min(std::min((obj_A.lower()*obj_B.lower()),(obj_A.lowe
160         r()*obj_B.upper())), std::min((obj_A.upper()*obj_B.lower()), (obj_A.upper()*obj_
161         B.upper())));
162         std::fesetround(FE_UPWARD);
163         upper_Temp = std::max(std::max((obj_A.lower()*obj_B.lower()),(obj_A.lowe
164         r()*obj_B.upper())), std::max((obj_A.upper()*obj_B.lower()), (obj_A.upper()*obj_
165         B.upper())));
166         interval<real_type> result(lower_Temp,upper_Temp);
167         result.incr_arithmetic_op_count();
168         return result;
169     }
170
171     template<class real_type>
172     bool operator<(const interval<real_type>& obj_A, const interval<real_type>&
173     obj_B){
174         if( (obj_A.upper()) < (obj_B.lower()) ) { return true; }

```



```
163         else if( (obj_A.lower()) >= (obj_B.upper()) ) { return false; }
164         else{
165             interval<real_type>::incr_indeterminate_result_count();
166             throw indeterminate_result("Indeterminate less than operator");
167         }
168     }
169
170     template<class real_type>
171     std::ostream& operator<<(std::ostream& outStream, const interval<real_type>&
172     obj) {
173         outStream<< "["<<(obj.lower())<<","<<(obj.upper())<<"]";
174         return outStream;
175     }
176
177 }
178 #endif
```

```
1  #include "ra/interval.hpp"
2  #include <iostream>
3
4  int main() {
5      using itvf = typename ra::math::interval<float>;
6      using stat = typename ra::math::interval<float>::statistics;
7      using std::cout;
8      using std::endl;
9
10     // Test default constructor
11     itvf obj_A;
12     cout<<"obj_A: "<<(obj_A.lower())<<" "<<(obj_A.upper())<<endl;
13     // Test 2 value constructor
14     itvf obj_B(2.89, 2.91);
15     itvf obj_C(3.91, 3.89);
16     cout<<"obj_B: "<<(obj_B.lower())<<" "<<(obj_B.upper())<<endl;
17     cout<<"obj_C: "<<(obj_C.lower())<<" "<<(obj_C.upper())<<endl;
18     // Test Compound add, sub, mult
19     cout<<"obj_C+=obj_B: "<<((obj_C+=obj_B).lower())<<" "<<(obj_C.upper())<<endl;
20     cout<<"obj_C-=obj_B: "<<((obj_C-=obj_B).lower())<<" "<<(obj_C.upper())<<endl;
21     cout<<"obj_C*=obj_B: "<<((obj_C*=obj_B).lower())<<" "<<(obj_C.upper())<<endl;
22     // Test Aliasing of Compound add, sub, mult
23     cout<<"obj_B+=obj_B: "<<((obj_B+=obj_B).lower())<<" "<<(obj_B.upper())<<endl;
24     cout<<"obj_B-=obj_B: "<<((obj_B-=obj_B).lower())<<" "<<(obj_B.upper())<<endl;
25     cout<<"obj_B*=obj_B: "<<((obj_B*=obj_B).lower())<<" "<<(obj_B.upper())<<endl;
26     // Test is_singleton
27     itvf obj_D(1.7);
28     cout<<"obj_D.is_singleton() must be true: "<<(obj_D.is_singleton())<<endl;
29     cout<<"obj_C.is_singleton() must be false: "<<(obj_C.is_singleton())<<endl;
30     // Test sign()
31     itvf obj_E(-7.335, -7.339);
32     itvf obj_F(3.442, 3.448);
33     itvf obj_G(-0.04, 0.12);
34     cout<<"Neg sign: "<<(obj_E.sign())<<endl;
35     cout<<"Pos sign: "<<(obj_F.sign())<<endl;
36     cout<<"Zero no sign: "<<(obj_A.sign())<<endl;
37     //cout<<"Indeterminate for sign: "<<(obj_G.sign())<<endl;
38     // Test get_statistics and clear_statistics
39     stat st;
40     obj_C.get_statistics(st);
41     cout<<"get_statistics of obj_C: indet="<<(st.indeterminate_result_count)<<" arith="<<(st.
arithmetric_op_count)<<endl;
42     obj_C.clear_statistics();
43     obj_C.get_statistics(st);
44     cout<<"clear_statistics of obj_C: indet="<<(st.indeterminate_result_count)<<" arith="<<(st.
arithmetric_op_count)<<endl;
45     // Test Binary add, sub, and mult
46     itvf obj_H = obj_E + obj_F;
47     itvf obj_I = obj_E - obj_F;
48     itvf obj_J = obj_E * obj_F;
49     cout<<"[-7.339,-7.335] + [3.442,3.448]: "<<obj_H<<endl;
50     cout<<"[-7.339,-7.335] - [3.442,3.448]: "<<obj_I<<endl;
51     cout<<"[-7.339,-7.335] * [3.442,3.448]: "<<obj_J<<endl;
52     // Test less than
53     itvf obj_L(-7.342, -7.337);
54     cout<<"Must be true: "<<(obj_E < obj_F)<<endl;
55     cout<<"Must be false: "<<(obj_A < obj_E)<<endl;
56     //cout<<"Indeterminate: "<<(obj_L < obj_E)<<endl;
57
58
59
60
```

```
61     return 0;  
62 }
```

```

1  #ifndef KERNEL_HPP
2  #define KERNEL_HPP
3  #include <CGAL/Cartesian.h>
4  #include <CGAL/MP_Float.h>
5  #include <cstdint>
6  #include "ra/interval.hpp"
7
8  namespace ra::geometry {
9      // A geometry kernel with robust predicates.
10     template<class R>
11     class Kernel {
12     public:
13         // The type used to represent real numbers.
14         using Real = R;
15
16         // The type used to represent points in two dimensions.
17         using Point = typename CGAL::Cartesian<R>::Point_2;
18
19         // The type used to represent vectors in two dimensions.
20         using Vector = typename CGAL::Cartesian<R>::Vector_2;
21
22         // Type used for interval class
23         using itv = typename ra::math::interval<R>;
24         // Type used for indeterminate_result class
25         using idr = typename ra::math::indeterminate_result;
26         // Type used for exact arithmetic
27         using exct = typename CGAL::MP_Float;
28
29         // The possible outcomes of an orientation test.
30         enum class Orientation : int {
31             right_turn = -1,
32             collinear = 0,
33             left_turn = 1,
34         };
35
36         // The possible outcomes of an oriented-side-of test.
37         enum class Oriented_side : int {
38             on_negative_side = -1,
39             on_boundary = 0,
40             on_positive_side = 1,
41         };
42
43         // The set of statistics maintained by the kernel.
44         struct Statistics {
45             // The total number of orientation tests.
46             std::size_t orientation_total_count;
47             // The number of orientation tests requiring exact arithmetic
48             std::size_t orientation_exact_count;
49             // The total number of preferred-direction tests.
50             std::size_t preferred_direction_total_count;
51             // The number of preferred-direction tests requiring exact arithmetic
52             std::size_t preferred_direction_exact_count;
53             // The total number of side-of-oriented-circle tests.
54             std::size_t side_of_oriented_circle_total_count;
55             // The number of side-of-oriented-circle tests requiring exact arithmetic
56             std::size_t side_of_oriented_circle_exact_count;
57         };
58
59         // Since a kernel object is stateless, construction and destruction
        are trivial

```

```

60         Kernel() = default;
61         ~Kernel() = default;
62
63         // The kernel type is both movable and copyable.
64         // Since a kernel object is stateless, a copy/move operation is trivial
65
66         Kernel(const Kernel&) = default;
67         Kernel& operator=(const Kernel&) = default;
68         Kernel(Kernel&) = default;
69         Kernel& operator=(Kernel&) = default;
70
71         // Member function to compute 3 by 3 determinant
72         template<class DR>
73         static DR determinant_3_by_3(const DR (&m)[3][3]) {
74             DR det = ( (m[0][0]*m[1][1]*m[2][2])+(m[0][1]*m[1][2]*m[2][0])+(
75                 m[0][2]*m[1][0]*m[2][1]) - ( (m[2][0]*m[1][1]*m[0][2])+(m[2][1]*m[1][2]*m[0][0]
76                 )+(m[2][2]*m[1][0]*m[0][1]) );
77             return det;
78         }
79
80         // Member function to compute 4 by 4 determinant
81         template<class DDR>
82         static DDR determinant_4_by_4(const DDR (&m)[4][4]) {
83             DDR sub0[3][3] = {m[1][1],m[1][2],m[1][3],m[2][1],m[2][2],m[2][3]
84                 ,m[3][1],m[3][2],m[3][3]};
85             DDR sub1[3][3] = {m[0][1],m[0][2],m[0][3],m[2][1],m[2][2],m[2][3]
86                 ,m[3][1],m[3][2],m[3][3]};
87             DDR sub2[3][3] = {m[0][1],m[0][2],m[0][3],m[1][1],m[1][2],m[1][3]
88                 ,m[3][1],m[3][2],m[3][3]};
89             DDR sub3[3][3] = {m[0][1],m[0][2],m[0][3],m[1][1],m[1][2],m[1][3]
90                 ,m[2][1],m[2][2],m[2][3]};
91             DDR det = (m[0][0]*determinant_3_by_3(sub0)) - (m[1][0]*determinant_3_by_3(sub1))
92                 + (m[2][0]*determinant_3_by_3(sub2)) - (m[3][0]*determinant_3_by_3(sub3));
93             return det;
94         }
95
96         // Determines how the point c is positioned relative to the
97         // directed line through the points a and b (in that order).
98         // Precondition: The points a and b have distinct values.
99         Orientation orientation(const Point& a, const Point& b, const Point&
100             c) const {
101             try{
102                 ++(stat_.orientation_total_count);
103                 itv ax(a.x()); itv ay(a.y()); itv bx(b.x()); itv by(b.y());
104                 itv cx(c.x()); itv cy(c.y());
105                 itv matrix[3][3] = {ax,bx,cx,ay,by,cy,itv(1.0),itv(1.0),itv(
106                     1.0)};
107                 itv det = determinant_3_by_3(matrix);
108                 if( (det.sign()) == -1 ) { return (Orientation::right_turn);
109             }
110             else if( (det.sign()) == 1 ) { return (Orientation::left_turn);
111         }
112             else{ return (Orientation::collinear); }
113         }
114         catch(const idr& e){
115             ++(stat_.orientation_exact_count);
116             exct matrix[3][3] = {(a.x()),(b.x()),(c.x()),(a.y()),(b.y()),
117                 (c.y()),exct(1.0),exct(1.0),exct(1.0)};
118             exct det = determinant_3_by_3(matrix);
119             if(det < exct(0)){ return (Orientation::right_turn); }
120             else if(det > exct(0)){ return (Orientation::left_turn); }
121         }

```

```

107         else{ return (Orientation::collinear); }
108     }
109 }
110
111 // Determines how the point d is positioned relative to the
112 // oriented circle passing through the points a, b, and c
113 // (in that order).
114 // Precondition: The points a, b, and c are not collinear.
115 Oriented_side side_of_oriented_circle(const Point& a, const Point& b
, const Point& c, const Point& d) const{
116     try{
117         ++(stat_.side_of_oriented_circle_total_count);
118         itv ax(a.x()); itv ay(a.y()); itv bx(b.x()); itv by(b.y());
119         itv cx(c.x()); itv cy(c.y());
120         itv dx(d.x()); itv dy(d.y());
121         itv ar((ax*ax)+(ay*ay)); itv br((bx*bx)+(by*by)); itv cr((cx
*cx)+(cy*cy));
122         itv dr((dx*dx)+(dy*dy));
123         itv matrix[4][4] = {ax,bx,cx,dx,ay,by,cy,dy,ar,br,cr,dr,itv(
1.0),itv(1.0),itv(1.0),itv(1.0)};
124         itv det = determinant_4_by_4(matrix);
125         if( (det.sign()) == -1 ) { return (Oriented_side::on_negativ
e_side); }
126         else if( (det.sign()) == 1 ) { return (Oriented_side::on_pos
itive_side); }
127         else{ return (Oriented_side::on_boundary); }
128     }
129     catch(const idr& e){
130         ++(stat_.side_of_oriented_circle_exact_count);
131         exct ax(a.x()); exct ay(a.y()); exct bx(b.x()); exct by(b.y(
));
132         exct cx(c.x()); exct cy(c.y());
133         exct dx(d.x()); exct dy(d.y());
134         exct ar((ax*ax)+(ay*ay)); exct br((bx*bx)+(by*by)); exct cr(
(cx*cx)+(cy*cy));
135         exct dr((dx*dx)+(dy*dy));
136         exct matrix[4][4] = {ax,bx,cx,dx,ay,by,cy,dy,ar,br,cr,dr,exc
t(1.0),exct(1.0),exct(1.0),exct(1.0)};
137         exct det = determinant_4_by_4(matrix);
138         if(det < exct(0)){ return (Oriented_side::on_negative_side);
}
139         else if(det > exct(0)){ return (Oriented_side::on_positive_s
ide); }
140         else{ return (Oriented_side::on_boundary); }
141     }
142 }
143
144 // Determines if, compared to the orientation of line
145 // segment cd, the orientation of the line segment ab is
146 // more close, equally close, or less close to the
147 // orientation of the vector v.
148 // The value returned is 1, 0, or -1 if, compared to the
149 // orientation of cd, the orientation of ab is more close,
150 // equally close, or less close to the orientation of v,
151 // respectively.
152 // Precondition: The points a and b have distinct values; the
153 // points c and d have distinct values; the vector v is not
154 // the zero vector.
155 int preferred_direction(const Point& a, const Point& b, const Point& c
, const Point& d, const Vector& v) const{
156     try{
157         ++(stat_.preferred_direction_total_count);

```

```

157         itv ax(a.x()); itv ay(a.y()); itv bx(b.x()); itv by(b.y());
158         itv cx(c.x()); itv cy(c.y()); itv dx(d.x()); itv dy(d.y());
159         itv vx(v.x()); itv vy(v.y());
160         itv first = ( (dx-cx)*(dx-cx) ) + ( (dy-cy)*(dy-cy) );
161         itv second_inc = ( (bx-ax)*(vx) ) + ( (by-ay)*(vy) );
162         itv second = second_inc * second_inc;
163         itv third = ( (bx-ax)*(bx-ax) ) + ( (by-ay)*(by-ay) );
164         itv fourth_inc = ( (dx-cx)*(vx) ) + ( (dy-cy)*(vy) );
165         itv fourth = fourth_inc * fourth_inc;
166         itv result = (first * second) - (third * fourth);
167         return (result.sign());
168     }
169     catch(const idr& e){
170         ++(stat_.preferred_direction_exact_count);
171         exct ax(a.x()); exct ay(a.y()); exct bx(b.x()); exct by(b.y(
172     ));
173         exct cx(c.x()); exct cy(c.y()); exct dx(d.x()); exct dy(d.y(
174     ));
175         exct vx(v.x()); exct vy(v.y());
176         exct first = ( (dx-cx)*(dx-cx) ) + ( (dy-cy)*(dy-cy) );
177         exct second_inc = ( (bx-ax)*(vx) ) + ( (by-ay)*(vy) );
178         exct second = second_inc * second_inc;
179         exct third = ( (bx-ax)*(bx-ax) ) + ( (by-ay)*(by-ay) );
180         exct fourth_inc = ( (dx-cx)*(vx) ) + ( (dy-cy)*(vy) );
181         exct fourth = fourth_inc * fourth_inc;
182         exct result = (first * second) - (third * fourth);
183         if( result < exct(0) ) { return -1; }
184         else if( result > exct(0) ) { return 1; }
185         else { return 0; }
186     }
187     }
188     // Tests if the quadrilateral with vertices a, b, c, and d
189     // specified in CCW order is strictly convex.
190     // Precondition: The vertices a, b, c, and d have distinct
191     // values and are specified in CCW order.
192     bool is_strictly_convex_quad(const Point& a, const Point& b, const Poi
193 nt& c, const Point& d) const {
194         Orientation t_1 = orientation(a,b,c);
195         Orientation t_2 = orientation(b,c,d);
196         Orientation t_3 = orientation(c,d,a);
197         Orientation t_4 = orientation(d,a,b);
198         //Orientation left = Orientation::left_turn;
199         //if( (t_1==left) && (t_2==left) && (t_3==left) && (t_4==left) )
200     { return true; }
201         if( (t_2==t_1) && (t_3==t_1) && (t_4==t_1) ){ return true; }
202         else{ return false; }
203     }
204     // Tests if the flippable edge, with endpoints a and c and
205     // two incident faces abc and acd, is locally Delaunay.
206     // Precondition: The points a, b, c, and d have distinct
207     // values; the quadrilateral abcd must be strictly convex.
208     bool is_locally_delaunay_edge(const Point& a, const Point& b, const Po
209 int& c, const Point& d) const{
210         Oriented_side tst = side_of_oriented_circle(a,b,c,d);
211         if(tst == Oriented_side::on_positive_side){ return false; }
212         else{ return true; }
213     }
214     // Tests if the flippable edge, with endpoints a and c and
215     // two incident faces abc and acd, has the preferred-directions

```

```

214         // locally-Delaunay property with respect to the first and
215         // second directions u and v.
216         // Precondition: The points a, b, c, and d have distinct values;
217         // the vectors u and v are not zero vectors; the vectors u and
218         // v are neither parallel nor orthogonal.
219         bool is_locally_pd_delaunay_edge(const Point& a, const Point& b, const
Point& c, const Point& d, const Vector& u, const Vector& v) const{
220             Oriented_side tst = side_of_oriented_circle(a,b,c,d);
221             if(tst == Oriented_side::on_positive_side){ return false; }
222             else if(tst == Oriented_side::on_negative_side){ return true; }
223             else{
224                 if((preferred_direction(a,c,b,d,u)) > 0) { return true; }
225                 else if( ((preferred_direction(a,c,b,d,u))==0) && ((preferre
d_direction(a,c,b,d,v))>0) ){
226                     return true;
227                 }
228                 else{ return false;}
229             }
230         }
231
232         // Clear (i.e., set to zero) all kernel statistics.
233         static void clear_statistics(){
234             stat_.orientation_total_count=0;
235             stat_.orientation_exact_count=0;
236             stat_.preferred_direction_total_count=0;
237             stat_.preferred_direction_exact_count=0;
238             stat_.side_of_oriented_circle_total_count=0;
239             stat_.side_of_oriented_circle_exact_count=0;
240         }
241
242         // Get the current values of the kernel statistics.
243         static void get_statistics(Statistics& statistics){
244             statistics = stat_;
245         }
246
247     private:
248         static Statistics stat_;
249
250 };
251
252     template<class Real>
253     typename Kernel<Real>::Statistics Kernel<Real>::stat_ = {.orientation_total_
count = 0, .orientation_exact_count = 0, .preferred_direction_total_count = 0, .
preferred_direction_exact_count = 0, .side_of_oriented_circle_total_count = 0, .
side_of_oriented_circle_exact_count = 0};
254
255
256 }
257 #endif
258

```



```

1  #include <CGAL/Kernel.h>
2  #include "ra/kernel.hpp"
3  #include <iostream>
4
5  int main() {
6      using std::cout;
7      using std::endl;
8      using knlD = typename ra::geometry::Kernel<double>;
9      using ort = typename ra::geometry::Kernel<double>::Orientation;
10     using ort_side = typename ra::geometry::Kernel<double>::Oriented_side;
11     using stat = typename ra::geometry::Kernel<double>::Statistics;
12     using point = typename CGAL::Cartesian<double>::Point_2;
13     using vector = typename CGAL::Cartesian<double>::Vector_2;
14
15     knlD obj_A;
16     point a(0,0); point b(2,2); point c(2,0); point d(1,1); point e(0,2);
17     // Test orientation
18     cout<<"Test for right(true): "<<((obj_A.orientation(a,b,c)) == (ort::right_turn))<<endl;
19     cout<<"Test for collinear(true): "<<((obj_A.orientation(a,b,d)) == (ort::collinear))<<endl;
20     cout<<"Test for left(true): "<<((obj_A.orientation(a,b,e)) == (ort::left_turn))<<endl;
21     // Test side_of_oriented_circle
22     point ac(0,0); point bc(2,0); point cc(0,2); point dc(1,1); point ec(2,2); point fc(3,3);
23     cout<<"Test for on_negative_side(true): "<<((obj_A.side_of_oriented_circle(ac,bc,cc,fc)) == (ort_side::on_negative_side))<<endl;
24     cout<<"Test for on_boundary(true): "<<((obj_A.side_of_oriented_circle(ac,bc,cc,ec)) == (ort_side::on_boundary))<<endl;
25     cout<<"Test for on_positive_side(true): "<<((obj_A.side_of_oriented_circle(ac,bc,cc,dc)) == (ort_side::on_positive_side))<<endl;
26     // Test preferred_direction
27     point ap(0,0); point bp(2,2); point cp(2,0); point dp(0,2);
28     vector up(2,1); vector vp(1,0); vector wp(-1,2);
29     cout<<"Test for more close(true): "<<((obj_A.preferred_direction(ap,bp,cp,dp,up)) == 1)<<endl;
30     cout<<"Test for equally close(true): "<<((obj_A.preferred_direction(ap,bp,cp,dp,vp)) == 0)<<endl;
31     cout<<"Test for less close(true): "<<((obj_A.preferred_direction(ap,bp,cp,dp,wp)) == -1)<<endl;
32     // Test is_strictly_convex_quad
33     point asc(0,0); point bsc(1,0); point csc(2,2); point dsc(1,2);
34     point anc(0,0); point bnc(1,0); point cnc(2,0); point dnc(2,1);
35     cout<<"Test for convexity(true): "<<((obj_A.is_strictly_convex_quad(asc,bsc,csc,dsc))<<endl;
36     cout<<"Test for convexity(false): "<<((obj_A.is_strictly_convex_quad(anc,bnc,cnc,dnc))<<endl;
37     // Test is_locally_delaunay_edge
38     point ad(0,0); point bd(1,0); point cd(1,1); point dd(0,1);
39     point adn(0,0); point bdn(5,0); point cdn(5,5); point ddn(2,3);
40     cout<<"Test edge for locally delaunay(true): "<<((obj_A.is_locally_delaunay_edge(ad,bd,cd,dd))<<endl;
41     cout<<"Test edge for locally delaunay(false): "<<((obj_A.is_locally_delaunay_edge(adn,bdn,cdn,ddn))<<endl;
42     // Test is_locally_pd_delaunay_edge
43     cout<<"Test edge for pref dir loc delaunay(true): "<<((obj_A.is_locally_pd_delaunay_edge(ad,bd,cd,dd,up,vp))<<endl;
44     cout<<"Test edge for pref dir loc delaunay(false): "<<((obj_A.is_locally_pd_delaunay_edge(ad,bd,cd,dd,wp,vp))<<endl;
45     // Test get_statistics and clear_statistics
46     stat st;

```

```
47     obj_A.get_statistics(st);
48     cout<<"Current statistics: ort tot="<<(st.orientation_total_count)<<" ort ex="<<(st.orientation_exact_count)<<" pd tot="<<(st.preferred_direction_total_count)<<" pd ex="<<(st.preferred_direction_exact_count)<<" crc tot="<<(st.side_of_oriented_circle_total_count)<<" crc ex="<<(st.side_of_oriented_circle_exact_count)<<endl;
49     obj_A.clear_statistics();
50     obj_A.get_statistics(st);
51     cout<<"Cleared statistics: ort tot="<<(st.orientation_total_count)<<" ort ex="<<(st.orientation_exact_count)<<" pd tot="<<(st.preferred_direction_total_count)<<" pd ex="<<(st.preferred_direction_exact_count)<<" crc tot="<<(st.side_of_oriented_circle_total_count)<<" crc ex="<<(st.side_of_oriented_circle_exact_count)<<endl;
52
53
54
55
56 }
```

```

1  #include<iostream>
2  #include"ra/kernel.hpp"
3  #include <CGAL/Cartesian.h>
4  #include "triangulation_2.hpp"
5  #include<vector>
6
7  using Kernel = CGAL::Cartesian<double>;
8  using kernel = ra::geometry::Kernel<double>;
9  using Triangulation = trilib::Triangulation_2<Kernel>;
10 using hEit = Triangulation::Halfedge_iterator;
11 using hE_hand = Triangulation::Halfedge_handle;
12 using point = CGAL::Cartesian<double>::Point_2;
13 using vector = CGAL::Cartesian<double>::Vector_2;
14
15 /*
16 using half_edge = Triangulation::Halfedge_handle;
17 using vertex = Triangulation::Vertex_handle;
18 */
19
20 bool is_flippable(hE_hand edg_iter, kernel& obj){
21     //if((*edg_iter).is_border_edge()){ return false; }
22     //else{
23         point a = edg_iter->vertex()->point();
24         point b = edg_iter->next()->vertex()->point();
25         point c = edg_iter->opposite()->vertex()->point();
26         point d = edg_iter->opposite()->next()->vertex()->point();
27         return (obj.is_strictly_convex_quad(a,b,c,d));
28     //}
29 }
30
31 int main(int argc, char** argv){
32     using std::cout;
33     using std::endl;
34     Triangulation tri(std::cin);
35     kernel obj;
36     hE_hand tmp_hand;
37
38
39     std::vector<hE_hand> suspect_list;
40     for (auto halfedgeIter = tri.halfedges_begin(); halfedgeIter !=tri.halfedges
_end(); ++halfedgeIter) {
41         tmp_hand = &*halfedgeIter;
42         if(is_flippable(tmp_hand,obj)){
43             suspect_list.push_back(tmp_hand);
44         }
45     }
46
47     while(!(suspect_list.empty())){
48         hE_hand it = suspect_list.back();
49         if(is_flippable(it,obj)){
50             point c = it->vertex()->point();
51             point d = it->next()->vertex()->point();
52             point a = it->opposite()->vertex()->point();
53             point b = it->opposite()->next()->vertex()->point();
54             if(!(obj.is_locally_pd_delaunay_edge(a,b,c,d,vector(1,0),vector(1,1)
55 ))){
56                 it = tri.flip_edge(it);
57                 suspect_list.push_back(it->next());
58                 suspect_list.push_back(it->next()->next());
59                 suspect_list.push_back(it->opposite()->next());
60                 suspect_list.push_back(it->opposite()->next()->next());
61             }

```

```

61         }
62         suspect_list.pop_back();
63     }
64
65     cout<<"Resulting Triangulation"<<endl;
66     tri.output_off(cout);
67
68     /*
69     int count = 0;
70     for (auto halfedgeIter = tri.halfedges_begin(); halfedgeIter !=tri.halfedges
_end(); +++halfedgeIter) {
71         if(is_flippable(halfedgeIter,obj)){ ++count; }
72         cout<<halfedgeIter->vertex()->point()<<endl;
73     }
74     cout<<"Count was: "<<count<<endl;
75     */
76
77
78
79     /*
80     vertex other_quad_pt_a; vertex other_quad_pt_b;
81     int num_of_edge_faces = 0;
82     // Iterate over every edge
83     for (auto halfedgeIter = tri.halfedges_begin(); halfedgeIter !=tri.halfedges
_end(); +++halfedgeIter) {
84         vertex he_a = halfedgeIter->vertex();
85         vertex he_b = halfedgeIter->opposite()->vertex();
86
87         // Check if that edge is flippable
88         // iterate over faces
89         for (auto faceIter = tri.faces_begin(); faceIter != tri.faces_end(); ++f
aceIter) {
90             half_edge h = faceIter->halfedge();
91             vertex face_a = h->vertex();
92             vertex face_b = h->opposite()->vertex();
93             vertex face_c = h->next()->vertex();
94             if( ((face_a==he_a)&&(face_b==he_b)) || ((face_a==he_b)&&(face_b==he
_a)) ) {
95                 ++num_of_edge_faces;
96                 if(num_of_edge_faces == 1){ other_quad_pt_a = face_c; }
97                 if(num_of_edge_faces == 2){ other_quad_pt_b = face_c; }
98             }
99             else if( ((face_a==he_a)&&(face_c==he_b)) || ((face_a==he_b)&&(face_
c==he_a)) ) {
100                 ++num_of_edge_faces;
101                 if(num_of_edge_faces == 1){ other_quad_pt_a = face_b; }
102                 if(num_of_edge_faces == 2){ other_quad_pt_b = face_b; }
103             }
104             else if( ((face_b==he_a)&&(face_c==he_b)) || ((face_b==he_b)&&(face_
c==he_a)) ) {
105                 ++num_of_edge_faces;
106                 if(num_of_edge_faces == 1){ other_quad_pt_a = face_a; }
107                 if(num_of_edge_faces == 2){ other_quad_pt_b = face_a; }
108             }
109             else{}
110         }
111     }
112     // If edge is attached to 2 faces, check if quad is convex (if edge is f
lippable)
113     if(num_of_edge_faces == 2){
114         if(obj.is_strictly_convex_quad((he_a.point()),(other_quad_pt_a.point
()),(he_b.point()),(other_quad_pt_b.point()))){

```

```
115         // It is flippable
116     }
117 }
118 num_of_edge_faces = 0;
119
120 }
121 */
122 }
123
124
125
126
127 /*
128 OFF
129 5 4 0
130 -1 -1 0
131 1 -1 0
132 1 1 0
133 -1 1 0
134 0 0 1
135 3 0 1 4
136 3 1 2 4
137 3 2 3 4
138 3 0 4 3
139 */
140
141
142 /*
143 OFF
144 9 8 0
145 -1 -1 0
146 0 -1 0
147 1 -1 0
148 1 0 0
149 1 1 0
150 0 1 0
151 -1 1 0
152 -1 0 0
153 0 0 0
154 3 0 1 8
155 3 0 8 7
156 3 1 2 3
157 3 1 3 8
158 3 3 4 8
159 3 8 4 5
160 3 8 5 7
161 3 7 5 6
162 */
```