

```
1 /*-----
2 Copyright (c) 2014 Author: Jagadeesh Vasudevamurthy
3 file: dsort.h
4 -----*/
5
6 /*-----
7 This file has dsort class declaration
8 -----*/
9
10 /*-----
11 All includes here
12 -----*/
13 #ifndef dsort_H
14 #define dsort_H
15
16 #include "../darray/darray.h"
17
18 /*-----
19 static definition - only once at the start
20 Change to false, if you don't need verbose
21 -----*/
22 template <typename T>
23 bool darray<T>::_display = false;
24
25 /*-----
26 class stat
27 -----*/
28 class statistics {
29 public:
30     statistics() :_nc(0), _ns(0), _ni(0) {}
31     void reset() {
32         _nc = 0; _ns = 0; _ni = 0;
33     }
34     void inc_num_compare() {
35         ++_nc;
36     }
37     void inc_num_swap() {
38         ++_ns;
39     }
40     void inc_num_iteration() {
41         ++_ni;
42     }
43     int get_num_compare() const {
44         return _nc;
45     }
46     int get_num_swap() const {
47         return _ns;
48     }
49     int get_num_iteration() const {
50         return _ni;
51     }
52 private:
53     int _nc; //num compare
54     int _ns; //num_swap
55     int _ni; //num_iteration
56 };
57
58 /*-----
59 class dsort
60 -----*/
61 template <typename T>
62 class dsort {
63 public:
64     dsort(darray<T> &d, int(*cf) (const T& c1, const T& c2), int size = 0);
65     ~dsort();
66     void bubble_sort();
```

```
67  bool binary_search(const T& r, int first, int last, int& middle);
68  void insertion_sort();
69  void merge_sort();
70  void quick_sort();
71  void change_swap_function(int(*cf) (const T& c1, const T& c2)) { _cf = cf; }
72  bool display()const { return _display; }
73  void set_size(int size) { _size = size; }
74  static void set_display(bool x) {
75      darray<T>::set_display(x);
76      _display = x;
77  }
78  void assertSorted();
79  //statistics
80  void reset_stat() {
81      _stat.reset();
82  }
83  void inc_num_compare() {
84      _stat.inc_num_compare();
85  }
86  void inc_num_swap() {
87      _stat.inc_num_swap();
88  }
89  void inc_num_iteration() {
90      _stat.inc_num_iteration();
91  }
92  int get_num_compare() const {
93      return _stat.get_num_compare();
94  }
95  int get_num_swap() const {
96      return _stat.get_num_swap();
97  }
98  int get_num_iteration() const {
99      return _stat.get_num_iteration();
100 }
101 /* no body will copies or equal dsort */
102 dsort(const dsort<T>& s) = delete;
103 dsort<T>& operator=(const dsort<T>& rhs) = delete;
104
105 private:
106     int _size;
107     statistics _stat;
108     darray<T>& _darray;
109     int(*_cf) (const T& c1, const T& c2);
110     static bool _display; /* ONLY ONCE for all object */
111
112     void _print_darray(int pass);
113     void _swap(T&a, T&b);
114     void _dump(int start, int end);
115 };
116
117 #include "dsort.hpp"
118
119 #endif
120 //EOF
121
122
```