**Response to Reviewers**

The authors thank the anonymous referee for his wonderful, constructive feedback. Below are our responses to the issues raised in the feedback.

1. For arrays of large items, we have used the third type, i.e., arrays of references. The references in our paper as well as in the implementation are the lists of numbers of length 16, 64, and 256, resp.  In fact, we don't know how to use the library function qsort() to sort a two-dimension array or an array of structured values.  Table 2 uses only lists of 64-bit integers. Details on arrays of various types are given in Tables 3-7. It is evident from TABLE 4 that hybrid methods are the best for lists of large items
2. Reference to *qsort* was updated and sentences such as "the best method" should be understood as "in our experiment".
3. In our experiment, -O3 is better than -O2 in the optimization of gcc. The change doesn't affect the ranking of methods.
4. URL references in the manuscript was converted to a reference.
5. The type of the array elements in Table 1 is integer. The manuscript was also updated.
6. A table of summary of C-library quick sort algorithm was created and referenced in the manuscript.
7. Latex issue and typos were all corrected.
8. The code has been changed so that no warnings will show up with gcc's option -Wall
9. We added a command line option "-R<n>" to replace script "repeat??", which allows sorting methods to work on <n> different instances. We added a command line option "-s" to replace script "runsubtype", which allows sorting methods to work on instances of various subtypes. We also added a command line option "-r<n>", which allows sorting methods to repeat <n> times on the same instance. This option is useful for testing methods on instances of small sizes. We will keep script "runtype" (which works on different types of instances), because creating an instance takes a lot of memory and we cannot reuse the memory if the type is changed. Of course, we may free the memory after each usage but we did not want garbage collection becomes a new issue.
10. We have modified makefile according to the reviewer’s suggestion. The default executable by the previous makefile is still optimied as defined by CFLAGS.
11. README file was updated based on reviewer’s suggestion.