

# problem set extra - 1

**Data Structures**  
**C++ for C Coders**

한동대학교 김영섭 교수  
idebtor@gmail.com

# Topics

- start from lab03.cpp → sort.cpp
- use new, delete, nothrow, and assert
- use a command line argument and GetInt() in nowic.h
  - using libnowic.a - a static library for getting user's input
  - -L../lib or -L./lib or -L../..lib
  - -lnowic, or lnnowic\_mac
- use a default parameter;
  - ex: void bubbleSort(int \*list, int N, int comp(int, int) = ascending);
- build a library libsort.a, sort.h
- use a function pointer to sort in either ascending or descending order
- use an array of function pointers for performance analysis
  - add timing as in pset02sort.
- **Due: March 18 (Mon) 11:55**
- **Files to submit: sort.cpp, sort.h, libsort.a, sort functions(4) in zipped**

# How to

step 1:

- start from lab03.cpp → sort.cpp
- get a copy of four sort files. set #if 1 for main() if necessary.
- build an executable.  
> g++ sort.cpp selection.cpp -o sort

step 2:

- read input from user from a command line and interactively.
- use new, delete, nothrow, and assert.
- build an executable.  
> g++ sort.cpp selection.cpp -I../../include -L../../lib -lnowic -o sort

step 3:

- code two comparison functions in sort.cpp
- define an array of function pointer and initialize with four sort functions.  
ex: void (\*fps[])(int \*, int, int (\*cmp)(int,int)) = {bubbleSort, insertionSort, ...};  
NOTE: Unfortunately, you cannot use a default argument in the function pointer declaration.
- add a for-loop such that it runs four sort functions one by one.
- build an executable.  
> g++ sort.cpp selection.cpp quicksort.cpp insertion.cpp bubble.cpp  
-I../../include -L../../lib -lnowic -o sort

[Note] don't go the next step unless your code above works completely.

# How to

step 4:

- code ascending and descending comparison functions in sort.cpp.
- add the ascending function as a default parameter in all four sort functions.
- add the comparison function proto-types in all four sort function files.
- build and test it.  
> g++ sort.cpp selection.cpp quicksort.cpp insertion.cpp bubble.cpp  
-I../../include -L../../lib -lnowic -o sort
- change all four sort functions to use the comparison function.
- add a comparison function pointer as an argument when you call a sort function.
- build and test it.

step 5:

- create include/sort.h.  
it should have a list of proto-types for four sort functions and two comparison functions
- use include guard #ifndef in sort.h. refer to nowic.h if necessary.
- use #include "sort.h" in sort.cpp and all four sort functions.
- remove sort function proto-types in sort.cpp
- remove comparison function proto-types in all four sort functions.
- build and test it.  
> g++ sort.cpp selection.cpp quicksort.cpp insertion.cpp bubble.cpp  
-I../../include -L../../lib -lnowic -o sort

# How to

step 6:

- Read `idebtor/nowic/UsingStaticLib.md`
- `libsort.a` using all four sort functions  
move `libsort.a` and `sort.h` where they belong to.
- build and test it using `lib/libsort.a` and `include/sort.h`  
> `g++ sort.cpp -I.././include -L.././lib -lnowic -lsort -o sort`

# Sample Run

```
Windows PowerShell
PS C:\GitHub\nowicx\psets\pset-extra1> ./sortx
Enter a number of samples: 10
Test Case: Not passing a default argument: quickSort(list, N)
-----
NOTsorted: 1 7 4 0 9 4 8 8 2 4
sorted: 0 1 2 4 4 4 7 8 8 9

Test Case: Passing a function pointer: quickSort(list, N, up)
-----
NOTsorted: 0 1 2 4 4 4 7 8 8 9
Up sorted: 0 1 2 4 4 4 7 8 8 9

Test Case: Passing a function pointer: quickSort(list, N, dn)
-----
NOTsorted: 0 1 2 4 4 4 7 8 8 9
Up sorted: 9 8 8 7 4 4 4 2 1 0

Test Case: Using an array of function pointers
-----
NOTsorted[Bubble]: 5 5 1 7 1 1 5 2 7 6
Up sorted[Bubble]: 1 1 1 2 5 5 5 6 7 7
Dn sorted[Bubble]: 7 7 6 5 5 5 2 1 1 1

NOTsorted[Insert]: 1 4 2 3 2 2 1 6 8 5
Up sorted[Insert]: 1 1 2 2 2 3 4 5 6 8
Dn sorted[Insert]: 8 6 5 4 3 2 2 2 1 1

NOTsorted[ Quick]: 7 6 1 8 9 2 7 9 5 4
Up sorted[ Quick]: 1 2 4 5 6 7 7 8 9 9
Dn sorted[ Quick]: 9 9 8 7 7 6 5 4 2 1

NOTsorted[Select]: 3 1 2 3 3 4 1 1 3 8
Up sorted[Select]: 1 1 1 2 3 3 3 3 4 8
Dn sorted[Select]: 8 4 3 3 3 3 2 1 1 1

Happy Coding~~
PS C:\GitHub\nowicx\psets\pset-extra1> .
```