

1. Write in C++ a non-templated iterative binary search algorithm which searches a vector of N integers for an integer key.
2. Write in C++ a template version of a recursive binary search function to search a vector. Specify requirements on the template parameter type. Discuss as comments the requirements on the template parameter type.
3. Now add to parts 1 & 2 above by filling the vector with N random integers, then searching the vector X times for a random integer in the range 0..N. N=10,000,000 and X = 10,000 Output the average search time (wall and CPU times) Note: you can use the type `unsigned long long int` for N
4. In addition to your programs for parts 1 and 2 above you need to write a short document giving your analyses of the algorithms, this analysis must be linked to the code in your program. Which implementation is faster, why etc.? To do this task you must collect running-time statistics for each search operation above.

To take CPU and wall times you will need library code. You may find the following links useful. <http://stackoverflow.com/questions/17432502/how-can-i-measure-cpu-time-and-wall-clock-time-on-both-linux-windows> If you wish to use the Boost API you will have to install it and set Visual Studio to access it. The getting started guide is here [https://www.boost.org/doc/libs/1\\_75\\_0/more/getting\\_started/windows.html](https://www.boost.org/doc/libs/1_75_0/more/getting_started/windows.html)

5. Create a C++ template function for the QuickSort algorithm. Do not change the function signature below. Discuss as comments the requirements on the template parameter type. (<http://en.wikipedia.org/wiki/Quicksort>)

```
template<typename T>
T* quicksort(T* array, int start, int end)
```