

Algorithms & Data Structures: Lab 12

week of 21st January 2019

1 Setup

1.1 Saving your work from last term

By now you should be familiar with the operations needed to save your work. Make sure you commit your work to version control often, and always have a backup copy, ideally remotely (for example in your own account on the department's gitlab installation.)

1.2 Downloading this week's distribution

Once you have successfully saved your changes from last week, you can get my updates by doing

```
git pull
```

which *should* automatically merge in new content. After the `git pull` command, you should have a new directory containing this week's material (named 12/) alongside your existing directories.

2 Binary search

Implement binary search on sorted arrays, as described in the lectures. You are provided with skeleton files and test files, as usual; running `make test` in the `cpp/` or `java/` directory should provide you with a test failure report, and once you have successfully implemented binary search, you should be able to rerun the tests with success.

Integrate an `OpCounter` to the class, to count comparisons between the search key and elements in the array that your code makes. Expose that functionality with the signature appropriate for your programming language:

```
Java int count(int array[], int key, int lo, int hi)
```

```
C++ int count(int *array, int key, size_t lo, size_t hi)
```

Are there (possibly invalid) combinations of arguments to your implementation of your search function which would lead to erroneous results or infinite loops? Can you write a test case to expose the bad behaviour? Send a pull request to the lab bundle with your new test.

2.1 Submission

Submit your work to the submission area for binary search. The submission system will remain open until **16:00 on Friday 25th January**; as usual, you may submit more than once, and your highest score is retained.