Sliding Puzzles Using C++ Report

Resources:

* <https://www.geeksforgeeks.org/unordered_map-in-cpp-stl/>
* <https://www.geeksforgeeks.org/pair-in-cpp-stl/>
* <https://www.geeksforgeeks.org/substring-in-cpp/>
* <https://www.geeksforgeeks.org/heap-using-stl-c/>
* <https://www.geeksforgeeks.org/implement-min-heap-using-stl/>
* <https://www.geeksforgeeks.org/unordered_set-in-cpp-stl/>
* <https://www.geeksforgeeks.org/measure-execution-time-function-cpp/>

Runtime Comparison - everything is run on my laptop for fair comparison

| Event | C++ (in seconds) | Python (in seconds) |
| --- | --- | --- |
| setting up 4-letter combos | 0.676 | 1.31 |
| solve the first 30 puzzles in 15puzzles.txt | 12.968 | 0.358 |

Report:

Surprisingly, my C++ program took significantly longer to run than my Python program. I believe this is because I had to write a custom class to encompass my data in my C++ program. In my A-Star code in Python, I inserted tuples of three elements into the queue. However, C++ doesn’t have a built in data structure for more than two elements, so I wrote a custom class that has three fields containing my data. In order for my priority queue to be able to contain this class (which I called Triple), I had to write a custom comparator function as well. Since I didn’t change anything else about the program, I assume that this is what slowed my C++ program down. This is further supported by the runtimes for the “setting up 4-letter combos” event, in which I copied the code word for word and the C++ program was much faster.

Learning C++ was not too bad, since it is really similar to Java. I also knew a little bit of C++ before doing this project because of the Computer Visions class I’m taking, but I didn’t (and still don’t) know much. C++ is probably the strictest programming language I have learned, so I got caught up in a lot of compiler errors while programming. I think that C++ is possibly the most unsuitable language for learning AI, because it is extremely not versatile and students would probably be too distracted by the confusing compiler errors instead of focusing on the actual AI algorithms. In addition, C++ does not support many data structures (such as tuples), and the ones that it does support are annoying to create (for example, unordered maps, which are basically dictionaries, are quite lengthy in their notation). The speed that C++ is supposed to have will be offset by the need to write custom classes anyway, so there won’t be many speed gains anyway. In conclusion, I discourage anyone from learning AI using C++.