

Course C++, Exercise Number 11

Date: 24.05.2012 + three weeks

1. Download **solve.h** and **solve.cpp** and make sure that you understand how they work. This means: Understanding the purpose of **cube_cmp**, of **struct pq**, and understanding how the function **solve** works. You must also understand how **std::map< >** is used in function **solve**.
2. Make sure that you can compile **solve.h** and **solve.cpp**. In order to do that, you need to add **cube::distance() const**
3. This task is optional: Currently, **solve** is not able to solve the cube that is constructed by **example3**. It fills up the memory, and throws a bad allocation error. What could be done about that?
 - Improve the distance function in **cube::distance()**.
 - When the memory is getting full, delete one third of the states with the biggest distance. Be careful, because deleted states must also be removed from the priority queue. Otherwise, there will be dangling map iterators.
 - In order to save memory, don't store the states of the cube. It is enough to store the sequences. The cube can always be reconstructed from the move sequence. This algorithm will run slower, but it be able to use much less memory, and therefore be able to get further in the search.