

Solving Quagmire 3 with a Keyspace algorithm.

Sometimes the method based on symmetry does not give enough information for you to solve. This can happen if the crib is very short and consequently there is insufficient information for symmetries to extend the plaintext and enable recognition of more words.

If that happens then the Keyspace algorithm may well find a solution provided that the keyword itself is brief – not more than 7 letters.

The idea of the keyspace method is that Hillclimbing is confined to the letters of the keyword rather than to the whole mixed alphabet. Thus if the keyword is 6 letters long, the Hillclimbing focuses on those 6 letters and the remaining letters are added in alphabetical order. The task is thus hugely reduced, even though the hillclimbing must be applied to each possible keyword length from (say) 5 to 7.

The Churn algorithm will give a faster result than Hillclimbing. Both algorithms are described in the section on the Churn algorithm.

Here is the algorithm for the Keyspace Hillclimber:

```
1.for keyspace from 5 to 11:
  2. make a keyword at random of length=keyspace;
  3. decipher and score, calling the score maxscore;
  4. do this cycle until no further improvement:
      -once in 5, swap two letters within the
        keyspace;
      -four times in 5, swap a letter in the
        keyspace with a letter outside and remake
        the letters following the keyspace in normal
        order;
      -decipher and score the plaintext;
      -if the score>maxscore
          {keep the key; make maxscore=score;}
      else
          swap back the letters to restore the
          unchanged key;
  when no further improvement from this cycle go
  back to (2), unless 10 million keys have been
  tried for the current keyspace in which case go
  back to (1)
```