Qixiang Liu
2856114
Mini project 1
02/13/19

Introduction:

The mini project implements a Vigenere Cipher. We are able to decode the first words of six messages in different keys.

Requirements:

When key length increases, cracking becomes slow. Please optimize your code, and at least finish the first 5.
Please record the time needed to decrypt each message.
Please submit your code and a short report. The report shall demonstrate the plaintexts that you discovered.
The report shall also discuss the performance (efficiency) of password cracking.

Result:

Text 1:
MSOKKJ COSXOEEKDTOSLGFWCMCHSUSGX (key length = 2 first word length = 6)
Key:KS
CAESARSWIFEMUSTBEABOVESUSPICION (CAESAR'S WIFE MUSTBE ABOVE SUSPICION)
Time: 0.566574 seconds.
After improving performance: 0.000891s

Text 2:
OOPCULN WFRCFQAQJGPNARMEYUODYOUNRGWORQEPVARCEPBBSCEQYEARAJUYGWWYACYWBPRNEJBMDTEAEYCCFJNENSGWAQRTSJTGXNRQRMDGFEEPHSJRGFCFMACCB (key length = 3 first word length = 7)
Key: JAY
FORTUNEWHICHHASAGREATDEALOFPOWERINOTHERMATTERSBUTESPECIALLYINWARCANBRINGABOUTGREATCHANGESINASITUATIONTHROUGHVERYSLIGHTFORCES
(FORTUNE WHICH HAS A GREAT DEAL OF POWER IN OTHER MATTERS BUT ESPECIALLY IN WAR CAN BRING ABOUT GREAT CHANGES IN A SITUATION THROUGH VERY SLIGHT FORCES –Julius Caesar)
Time: 23.806740 seconds.
After improving performance: 0.014939s

Text 3:
MTZHZEOQKA SVBDOWMWMKMNYIIHVWPEXJA (key length = 4 first word length = 10)
Key: IWKD
EXPERIENCEISTHETEACHEROFALLTHINGS (EXPERIENCE IS THE TEACHER OF ALL THINGS)

Time: 641.628296 seconds. (about 10min) Slow!!
After improving performance: 0.461201s

Text 4:
HUETNMIXVTM QWZTQMMZUNZXNSSBLNSJVSJQDLKR(Key length = 5 first word length = 11)
Key: ZIENF
IMAGINATIONISMOREIMPORTANTTHANKNOWLEDGE.
(IMAGINATION IS MORE IMPORTANT THAN KNOWLEDGE.)
Time: 3135.416260 seconds. (about 52min)
After improving performance: 11.717607s
Text 5:
LDWMEKPOP
SWNOAVBIDHIPCEWAETYRVOAUPSINOVDIEDHCDSELHCCPVHRPOHZUSERSFS
(key length =6 first word length = 9)
Key: HACKER
EDUCATIONISWHATREMAINSAFTERONEHASFORGOTTENWHATONEHASLEARNEDINSCHOOL
(EDUCATIONs IS WHAT REMAINS AFTER ONE HAS FORGOTTEN WHAT ONE HAS LEARNED IN SCHOOL –Albert Einstein)
After Time: 385.714844s

Text 6:
VVVLZWWPBWHZDKBTXLDCGOTGTGRWAQWZSDHEMXLBELUMO
Key:NICHOLS
INTELLECTUALSSOLVEPROBLEMSGENIUSESPREVENTTHEM (INTELLECTUALS SOLVE PROBLEMS GENIUSES PREVENT THEM –Albert Einstein)
After Time: 7860.297852 (about 2.1h)

Discussion:

How to improve performance:

1. Only decrypt the first number of words, so I can split ciphertexts into two parts. If I get the key from the first word, the left part is easy to be gotten.
2. Because of known the length of the first word, I can split the dictionary into different lengths in order to check results.
3. Key length decides how many cycles of decryption, eg AAA-ZZZ or AAAA-ZZZZ;
4. The algorithm is about decryption method. It also affects performance strongly. I use to subtract between characters. The way is so bad because I need consider negative or positive ASCII code. I suggest using index to represent 26 letters.
5. Check if the plaintext is correct. It is related what is type of data structure we have in the dictionary.

Hint: Array cannot has size of 26^7 in c++ in my computer.

Conclusion:

Im conclusion, when I did do search something in dictionary. I should consider what data structure can find correct result quickly. In the beginning, I used the vector to store dictionary, the the worst performance is Big-O(n), so I should change data structure to hash table. Moreover, I should consider which data need to be stored because store data decrease performance. At start, I store all keys in different key lengths. Finally, I found it cannot store all keys because 26^7 is a big number. Therefore, I need to use once replace once in recursion. The algorithm is about decryption that also can improve performance. I think ASCII addition and subtraction are slower than integer addition and subtraction, but I did not test the field. There are two texts from Einstein.