REPORT ON CRYPTOGRAPHY

September 17, 2019

Ahsan Aziz Ishan
Shahjalal University of Science & Technology
Department of Software Engineering

Contents

0.1 Encrypted Message

aceah toz puvg vcdl omj puvg yudqecov, omj loj auum klu thmjuv hs klu zlcvu shv zcbkg guovz, upuv zcmdu lcz vuwovroaeu jczoyyuovomdu omj qmubyudkuj vukqvm. klu vcdluz lu loj avhqnlk aodr svhw lcz kvopuez loj mht audhwu o ehdoe eunumj, omj ck toz yhyqeoveg auecupuj, tlokupuv klu hej sher wcnlk zog, klok klu lcee ok aon umj toz sqee hs kqmmuez zkqssuj tckl kvuozqvu. omj cs klok toz mhk umhqnl shv sowu, kluvu toz oezh lcz yvhehmnuj pcnhqv kh wovpue ok. kcwu thvu hm, aqk ck zuuwuj kh lopu eckkeu ussudk hm wv. aonncmz. ok mcmukg lu toz wqdl klu zowu oz ok scskg. ok mcmukg-mcmu klug aunom kh doee lcw tuee-yvuzuvpuj; aqk qmdlomnuj thqej lopu auum muovuv klu wovr. kluvu tuvu zhwu klok zlhhr klucv luojz omj klhqnlk klcz toz khh wqdl hs o nhhj klcmn; ck zuuwuj qmsocv klok omghmu zlhqej yhzzuzz (oyyovumkeg) yuvyukqoe ghqkl oz tuee oz (vuyqkujeg) cmubloqzkcaeu tuoekl. ck tcee lopu kh au yocj shv, klug zocj. ck czm'k mokqvoe, omj kvhqaeu tcee dhwu hs ck! aqk zh sov kvhqaeu loj mhk dhwu; omj oz wv. aonncmz toz numuvhqz tckl lcz whmug, whzk yuhyeu tuvu tceecmn kh shvncpu lcw lcz hjjckcuz omj lcz nhhj shvkqmu. lu vuwocmuj hm pczckcmn kuvwz tckl lcz vueokcpuz (ubduyk, hs dhqvzu, klu zodrpceeuaonncmzuz), omj lu loj womg juphkuj ojwcvuvz owhmn klu lhaackz hs yhhv omj qmcwyhvkomk sowcecuz. aqk lu loj mh dehzu svcumjz, qmkce zhwu hs lcz ghqmnuv dhqzcmz aunom kh nvht qy. klu uejuzk hs kluzu, omj aceah'z sophqvcku, toz ghqmn svhjh aonncmz. tlum aceah toz mcmukg-mcmu lu ojhykuj svhjh oz lcz lucv, omj avhqnlk lcw kh ecpu ok aon umj; omj klu lhyuz hs klu zodrpceeu- aonncmzuz tuvu scmoeeg jozluj. aceah omj svhjh loyyumuj kh lopu klu zowu acvkljog, zuykuwauv 22mj. ghq loj aukkuv dhwu omj ecpu luvu, svhjh wg eoj, zocj aceah hmu jog; omj klum tu dom dueuavoku hqv acvkljog-yovkcuz dhwshvkoaeg khnukluv. ok klok kcwu svhjh toz zkcee cm lcz ktuumz, oz klu lhaackz doeeuj klu cvvuzyhmzcaeu ktumkcuz auktuum dlcejlhhj omj dhwcmn hs onu ok klcvkg-klvuu

Frequency distribution English characters

a:	8.05%	b:	1.67%	c:	2.23%	d:	5.10%
e:	12.22%	f:	2.14%	g:	2.30%	h:	6.62%
i:	6.28%	j:	0.19%	k:	0.95%	l:	4.08%
m:	2.33%	n:	6.95%	o:	7.63%	p:	1.66%
q:	0.06%	r:	5.29%	s:	6.02%	t:	9.67%
u:	2.92%	v:	0.82%	w:	2.60%	x:	0.11%
y:	2.04%	z:	0.06%				

Figure 1: Frequency Table

0.2 Finding a Solution

The encrypted text could be a English paragraph. So, our first assumption is the paragraph is written in English. To decrypt the text, firstly we need to identify some especial characters of English alphabet which occurs more than other alphabets. Then we will approach in an iterative way of making assumptions of some words and trying to find out more characters what they mean. For this we will be using a frequency table to identify some most frequent characters.

Figure 1 shows a frequency distribution of english letters.

0.3 The Process

1. Firsty we wrote a simple **C++** code to count the occurances of different characters.
2. From the output in Table 1 we found out that most used character in this paragraph is **u**.
3. As the most frequently appeared English letter is **e**, we assume **u** means **e**.
4. Now from frequency distribution and our data in Table 1 we can replace most common first 6 or 7 letters and search some predictable words again in the document.
5. So,we replace
   u -> **e**,
   k -> **t**
   o -> **a**,
   h -> **o**,
   c -> **n**,
   z -> **s**
6. After these operations we go through the text here.

aneao tas **pevg** vndh amj pevg yedqenav, amj haj aeem the tomjev os the shnve sov snbtg geavs, epev snmde hns vewavraaee jnsayyeavamde amj qmebyedtej vetqvm. the vndhes he haj avoqnht aadr svow hns tvapees haj mot aedowe a eodae eenemj, amj nt tas yoyqeaveg aeenepej, thatepev the oej soer wnnht sag, that the hnee at aan emj tas sqee os tqmmees stqssej tnth tveasqve. amj ns that tas mot emoqnh sov sawe, theve tas aeso hns yvoeomnej pnnoqv to wavpee at. tnwe tove om, aqt nt seewej to hape enttee essedt om wv. aannnms. at mnmetg he tas wqdh the sawe as at snstg. at mnmetg-mnme theg aenam to daee hnw teee-yvesevpej; aqt qmdhamnej toqej hape aeem meavev the wavr. theve **teve** sowe that shoor thenv heajs amj thoqnht thns tas too wqdh os a nooj thnmn; nt seewej qmsanv that amgome shoqej yossess (ayyavemteg) yevyetqae goqth as teee as (veyqtejeg) nmebhaqstnaee teaeth. nt tnee hape to ae yanj sov, theg sanj. nt nsm't matqvae, amj tvoqaee tnee dowe os nt! aqt so sav tvoqaee haj mot dowe; amj as wv. aannnms tas nemevoqs tnth hns womeg, wost yeoyee teve tneenmn to sovnnpe hnw hns ojjntnes amj hns nooj sovtqme. he vewanmej om pnsntnmn tevws tnth hns veeatnpes (ebdeyt, os doqvse, the sadrpneeeaannnmses), amj he haj wamg jepotej ajwnvevs awomn the hoaants os yoov amj qmnwyovtamt sawnenes. aqt he haj mo deose svnemjs, qmtne sowe os hns goqmnev doqsnms aenam to nvot qy. the **eejest os** these, amj aneao's sapoqvnte, tas goqmn svojo aannnms. them aneao tas mnmetg-mnme he ajoytej svojo as hns henv, amj avoqnht hnw to enpe at aan emj; amj the hoyes os the sadrpneee- aannnmses teve snmaeeg jashej. aneao amj svojo hayyemej to hape the sawe anvthjag, seytewaev **22mj**. goq haj aettev dowe amj enpe heve, svojo wg **eaj**, sanj aneao ome jag; amj them te dam deeeavate oqv anvthjag-yavtnes dowsovtaaeg tonethev. at that tnwe svojo tas stnee nm hns tteems, as the hoaants daeeej the nvvesyomsnaee ttemtnes aetteem dhnejhooj amj downmn os **ane** at **thnvtg-thvee**

7. If we observe closely now, we can see that there are some predictable words, which can be used to find out more words. Such as, the word **thnvtg-thvee** most probably means **thirty-three** and **22mj** means **22nd**.

So, we get
n->i
j->d
v->r
g->y

8. Now, we can use our assumed knowledge to find out more meaningful words, from **eaj** we assume its **lad**, and **perg** is **very**

9. So we get again,
p->v
e->l

10. On next iteration, **teve** is **were** **eejest** assumed as **eldest**, **os** is **of** and **ane** is **age** So we get,
e->l
t->w
s->f
n->g
*conflict

11. With the concept of, **Top-Sort Algorithm** we can form a tree-like-graph to define which character should be replaced before which, otherwise it will lead to conflict. So, We form a graph (see figure 2) finally after finding out some more words and replace the letters starting from the root of the tree. The iterative approach continues everytime from begin like this.

12. As we can see, here also some conflicts arise too, so we go back to our initial assumption and **change the fact c->n which later became n->i by directly assuming c->i** and start iterations again as decrypting is no easy-stuff.

u	198	k	132	o	131	h	113
c	102	l	97	z	95	m	95
v	85	j	74	e	71	a	47
q	42	w	38	s	38	n	37
t	34	d	29	g	28	y	28
p	22	r	7	b	4	2	2

Table 1: Frequency distribution of Encrypted letters

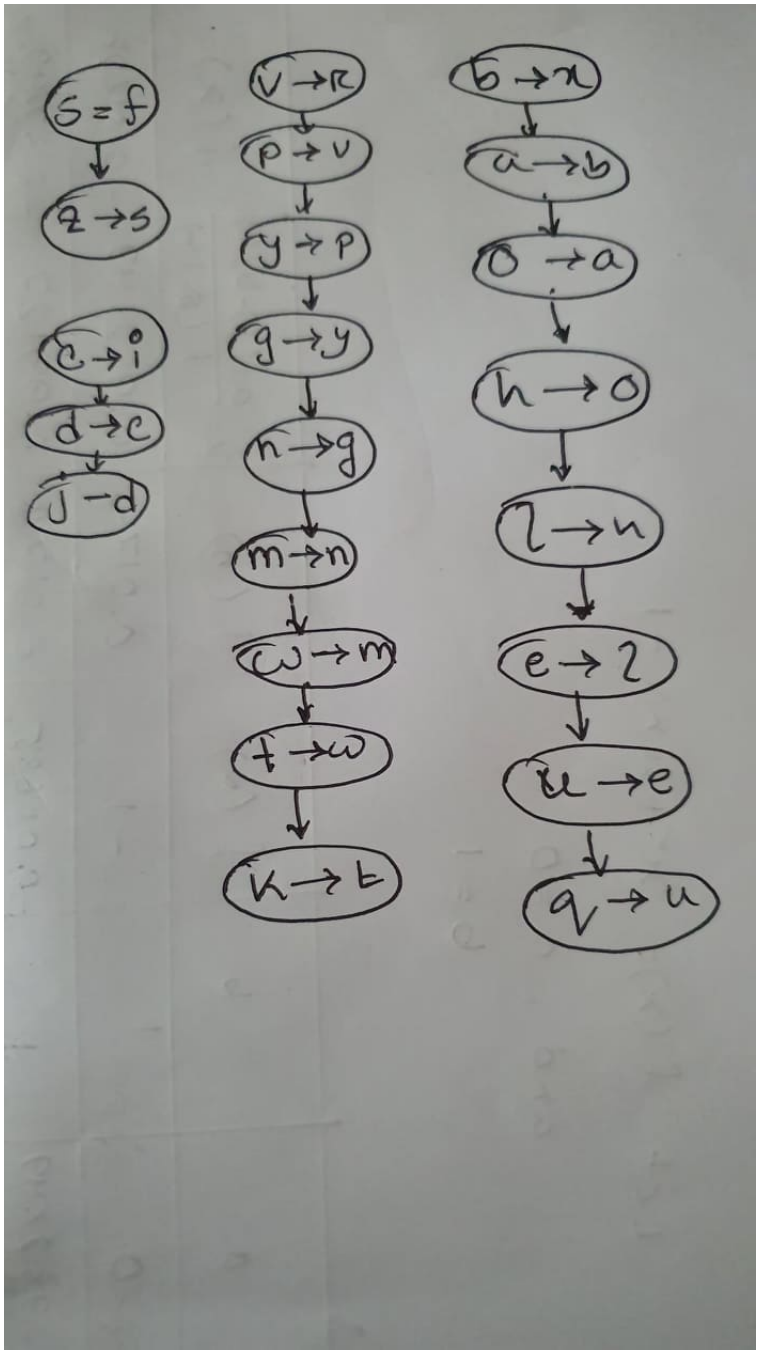

Figure 2: Graph of Iteration order

0.4 Final readable text

bilbo was very rich and very peculiar, and had been the wonder of the shire for sixty years, ever since his remarkable disappearance and unexpected return. the riches he had brought back from his travels had now become a local legend, and it was popularly believed, whatever the old folk might say, that the hill at bag end was full of tunnels stuffed with treasure. and if that was not enough for fame, there was also his prolonged vigour to marvel at. time wore on, but it seemed to have little effect on mr. baggins. at ninety he was much the same as at fifty. at ninety-nine they began to call him well-vreserved; but unchanged would have been nearer the mark. there were some that shook their heads and thought this was too much of a good thing; it seemed unfair that anyone should possess (apparently) pervetual youth as well as (revutedly) inexhaustible wealth. it will have to be vaid for, they said. it isn't natural, and trouble will come of it! but so far trouble had not come; and as mr. baggins was generous with his money, most veovle were willing to forgive him his oddities and his good fortune. he remained on visiting terms with his relatives (excevt, of course, the sackvillebagginses), and he had many devoted admirers among the hobbits of poor and unimportant families. but he had no close friends, until some of his younger cousins began to grow up. the eldest of these, and bilbo's favourite, was young frodo baggins. when bilbo was ninety-nine he adopted frodo as his heir, and brought him to live at bag end; and the hopes of the sackville- bagginses were finally dashed. bilbo and frodo happened to have the same birthday, september 22nd. you had better come and live here, frodo my lad, said bilbo one day; and then we can celebrate our birthday-parties comfortably together. at that time frodo was still in his tweens, as the hobbits called the irresponsible twenties between childhood and coming of age at thirty-three

0.5 Comments and Assumptions

The process is too much time consuming and needs to check the paragraph again and again Red color is used in 2nd iteration, Blue color is used in 3rd iteration and Green color is used in 4th iteration in the document.The C++ code to count words is given

below:

```
#include<bits/stdc++.h>
using namespace std;

map<char,int> m;

int main()
{
    char s;
    double d;
    while(cin>>s) {
        m[s]++;
    }
    for(auto it = m.begin();it != m.end(); ++it) {
        cout << it->first << " " << it->second << endl;
    }
    return 0;
}
```