Research Topic 8: Locks and letters

2018-2019

1 Introduction

Students: Andreica Amos[1], Balea Patricia[1], Căinap Maria[1], Cîmpean Flavia[1], Păcurar Flavia[1], Pașca Teodora[1], Renau Julie[2], Robin Bastien[2], Rouanet Emmie[2], Rus Carina[1], Salle Evane[2], Turcan Mihail[1]
Teacher: Ariana Văcărețu[1]
Researcher: Lorand Parajdi

2 Task

We want to create the most secure combination padlock with code we possibly can. We assume that most users would probably use common English words as the opening code.

How should we choose the number of dials, the number of letters per dial and the letters to write on each dial in order to maximize the number of combinations and thus render a thief's task much more difficult?

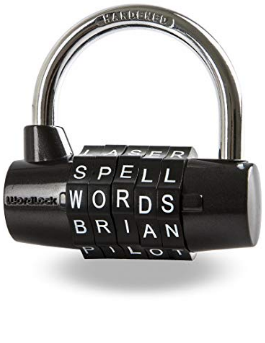


Figure 1: An image of a lock with letters instead of numbers

3 Methods

Supposing the code can be made up of any given combination of the letters of the English alphabet (26 in total), then the number of possibilities is as follows:

$$\binom{n}{26} \cdot n^m$$

Where m is the number of dials and n the number of letters per dial.

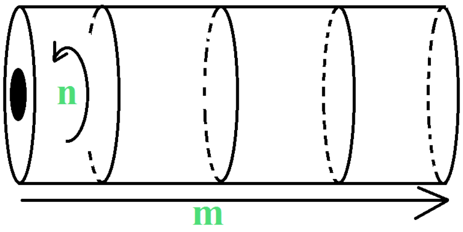


Figure 2: A graphic representation of the lock

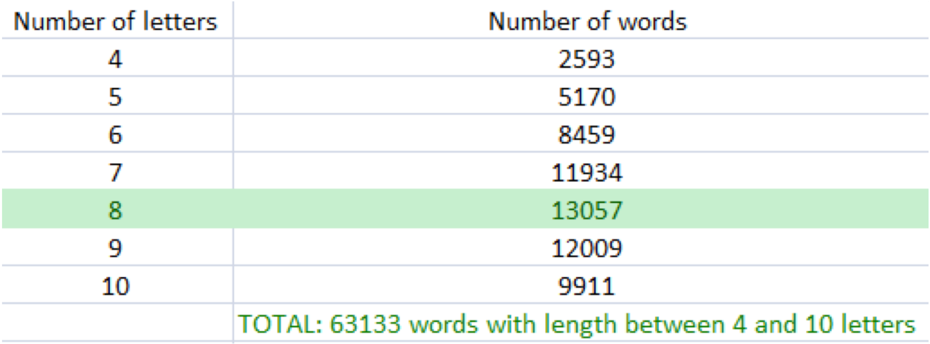
4 Requirements

Given the constraint – all passwords must be existent English words – and assuming the thief is aware of this particular rule, the end result depends on the database we choose our words from.

5 Direct approach

The first difficulty we encountered in the process of solving the problem was the choice of a common database regarding language, as the same English vocabulary was supposed to be used by both our team and the French team. After a discussion with the French researcher who proposed the problem, we were advised to try to solve it by using an online software program (the program can be found on the link: http://www.informathiques.fr /resources/exploredico.html). It features a 92518-word English-language vocabulary and gives the user the possibility to find out how many words there are of a certain form he or she has introduced. For example, if we want to find the number of words containing 5 letters and starting with the group ”ab” and the last letter being “s”, we will enter the word ”ab ** s” where a, b, s are letters on fixed positions and ”*” represent the positions on which the computer will put alphabet letters to get existing English words. It is important to specify that when entering the word ”ab ** s” the computer will return the figure ”5” representing the number of 5-letter English words out of the total of 92518 words. These 5 words are: abbes, abets, ables, abuts, abyss.

Having solved this first issue, we decided to start work on the problem per se. As we specified in "Task", the question we were supposed to provide an answer to was the following: "How should we choose the number of dials, the number of letters per dial and the letters to write on each dial in order to maximize the number of combinations and thus render a thief's task much more difficult?". To begin with, we had to find out which was the number of dials required in the construction of the lock in order to maximize the number of combinations. This figure represented the actual number of letters that the "keyword" might have. For instance, if our lock had had 4 dials, then obviously we could have created only four-letter words. But if our purpose was to make it almost impossible for the thief to guess the "keyword", then the number of possible words to be formed by the separate rotation of the dials should have been as large as possible. Therefore, we decided to introduce words of the "**..*" pattern in the program, one at a time, in order to gradually discover how many of the 92518 words were formed of 4 to 10 letters. The process we settled for was a gradual one: first we found out how many 4-letter words there were, then 5, 6 and so on. The following chart presents the obtained results.



Number of letters	Number of words
4	2593
5	5170
6	8459
7	11934
8	13057
9	12009
10	9911
	TOTAL: 63133 words with length between 4 and 10 letters

Figure 3: The number of words depending on the number of letters

6 The first results

By analysing the data we obtained by the method described above, it becomes clear that most words in our vocabulary contain 8 letters. In conclusion, by using a lock that contains 8 dials we will manage to create the highest number of words that can be formed. This number is of 13057 words and would only be reached when the 8 dials of the lock contain the entire alphabet used in the English language, with all its 26 letters. Only by using all these letters will we be able to form the total of 13057 8-letter words. Therefore, we thus provide an answer to the second part of the question as well. In order to create the most secure version of the padlock, the number of letters on each dial must be of 26 and the letters displayed on the dial should contain the entire alphabet. This is how we managed to find the solution to the problem after carrying out some specific research.

7 Further Considerations

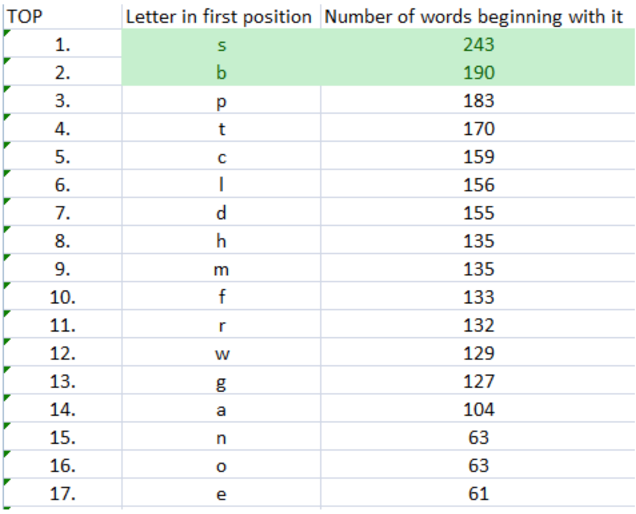
Naturally, the research does not stop at this point. Once we obtained the answer to the problem-proper, we decided to extend the research and determine the letters which should be written on the dials considering how many dials and how many letters per dial we take into account. It is important to specify that this approach to the problem is a practical one. Because a lock with 26 letters per dial would prove to be somewhat impractical if not impossible to manufacture, we have decided to reduce the problem to a more efficient, user-friendly padlock. As a result, we thought that for a particular given task (example: A client wants to have a padlock created with only 4 dials and 2 letters on each dial), we would find out what letters should be put on each dial. In the attempt to solve this problem we have put forward an algorithm.

8 The algorithm

The algorithm consists of a series of steps by which we would start a procedure, stemming from the given number of dials and the one of letters per dial, in order to find out which letters should be placed on each dial by using the software sent to us by the French researcher. In order to achieve a clearer understanding of the algorithm we opted for the concrete variant. For example, we chose a 4 dials padlock with 2 letters on each dial.

The first step consisted in determining the letters on the first dial (the letters which can represent the first letter in the “keyword”). These letters can be easily found by gradually introducing the words of the ”a***”, “b***”, “c***” (so on and so forth) pattern. The word whose introduction would return the largest number would represent the most efficient letter to be placed on the first position.

Therefore, by working on the concrete example, the letters on the first position are “b” and “s”, as can be observed from the chart.

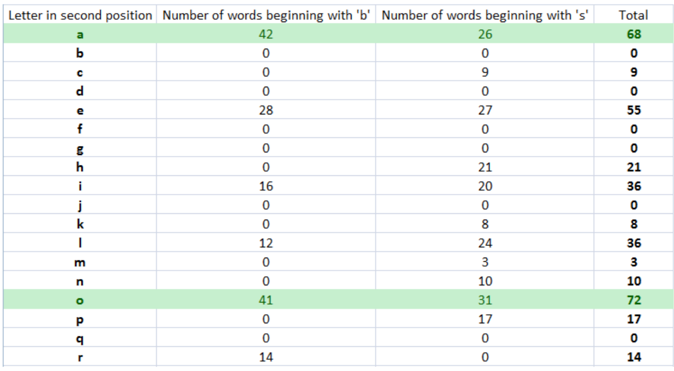


TOP	Letter in first position	Number of words beginning with it
1.	s	243
2.	b	190
3.	p	183
4.	t	170
5.	c	159
6.	l	156
7.	d	155
8.	h	135
9.	m	135
10.	f	133
11.	r	132
12.	w	129
13.	g	127
14.	a	104
15.	n	63
16.	o	63
17.	e	61

Figure 4: The number of words depending on the letter on the first position

After the most efficient two letters for the first position were determined we would move on to the second position. Determining the next two letters for the second position becomes a more complex and challenging task. We can no longer determine the most efficient letters by introducing the ”*a**“, ”*b**“, ”*c**“ etc. pattern words. Instead, we will have to consider both the letter on the first position as well as the one on the second position. As a result, we will introduce the words of the ”ba**”, ”bb**”, and so forth, and respectively ”sa**”, ”sb**” etc. pattern and we will monitor which would be the first two letters that could form the most 4 letter words, having one of the letters ”b” and “s“.

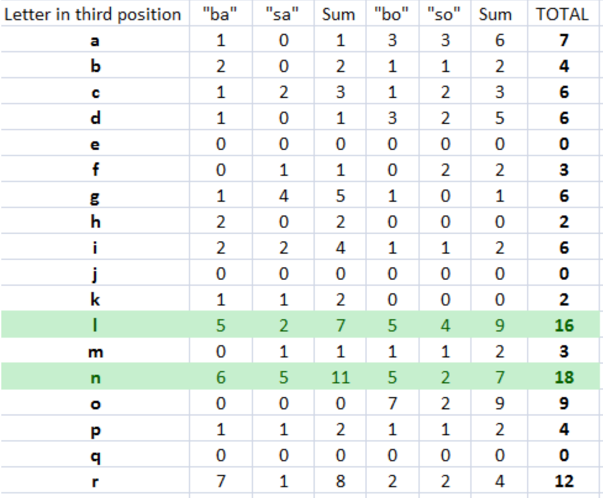
The result of this procedure would be that the second position should feature the letters “a” or “o”.



Letter in second position	Number of words beginning with 'b'	Number of words beginning with 's'	Total
a	42	26	**68**
b	0	0	**0**
c	0	9	**9**
d	0	0	**0**
e	28	27	**55**
f	0	0	**0**
g	0	0	**0**
h	0	21	**21**
i	16	20	**36**
j	0	0	**0**
k	0	8	**8**
l	12	24	**36**
m	0	3	**3**
n	0	10	**10**
o	41	31	**72**
p	0	17	**17**
q	0	0	**0**
r	14	0	**14**

Figure 5: The number of words depending on the letters on the first and second position

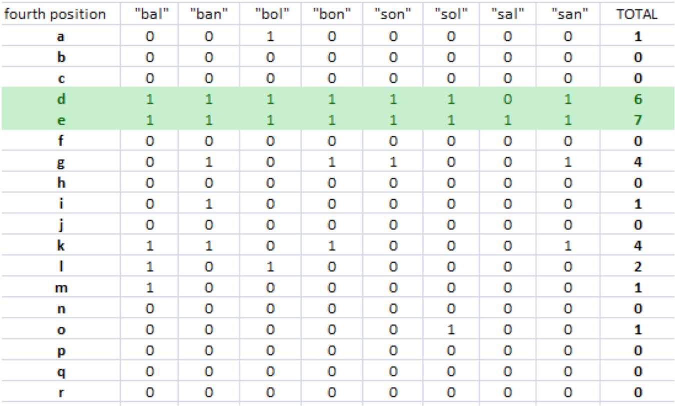
For the letters on the 3rd and 4th positions, we would apply the same procedure. The result would be that the third position should feature the letters “l” or “n”.



Letter in third position	"ba"	"sa"	Sum	"bo"	"so"	Sum	TOTAL
a	1	0	1	3	3	6	**7**
b	2	0	2	1	1	2	**4**
c	1	2	3	1	2	3	**6**
d	1	0	1	3	2	5	**6**
e	0	0	0	0	0	0	**0**
f	0	1	1	0	2	2	**3**
g	1	4	5	1	0	1	**6**
h	2	0	2	0	0	0	**2**
i	2	2	4	1	1	2	**6**
j	0	0	0	0	0	0	**0**
k	1	1	2	0	0	0	**2**
l	5	2	7	5	4	9	**16**
m	0	1	1	1	1	2	**3**
n	6	5	11	5	2	7	**18**
o	0	0	0	7	2	9	**9**
p	1	1	2	1	1	2	**4**
q	0	0	0	0	0	0	**0**
r	7	1	8	2	2	4	**12**

Figure 6: The number of words depending on the letter on the third position

For the last position, we have ‘d’ and ‘e’.



fourth position	"bal"	"ban"	"bol"	"bon"	"son"	"sol"	"sal"	"san"	TOTAL
a	0	0	1	0	0	0	0	0	1
b	0	0	0	0	0	0	0	0	0
c	0	0	0	0	0	0	0	0	0
d	1	1	1	1	1	1	0	1	6
e	1	1	1	1	1	1	1	1	7
f	0	0	0	0	0	0	0	0	0
g	0	1	0	1	1	0	0	1	4
h	0	0	0	0	0	0	0	0	0
i	0	1	0	0	0	0	0	0	1
j	0	0	0	0	0	0	0	0	0
k	1	1	0	1	0	0	0	1	4
l	1	0	1	0	0	0	0	0	2
m	1	0	0	0	0	0	0	0	1
n	0	0	0	0	0	0	0	0	0
o	0	0	0	0	0	1	0	0	1
p	0	0	0	0	0	0	0	0	0
q	0	0	0	0	0	0	0	0	0
r	0	0	0	0	0	0	0	0	0

Figure 7: The number of words depending on the letter on the last position

For the specified criteria, we have ‘b’ and ‘s’ on the first dial, ‘a’ and ‘o’ on the second, ‘l’ and ‘n’ on the third and ‘e’ and ‘d’ on the last. A possible combination for the above mentioned case might be:

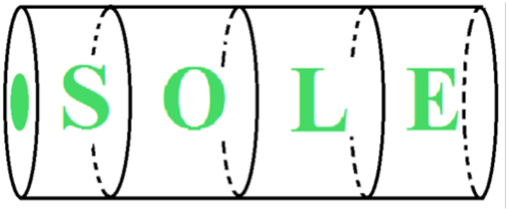


Figure 8: A possible combination for a padlock with 4 dials and 2 letters per dial

9 The French approach

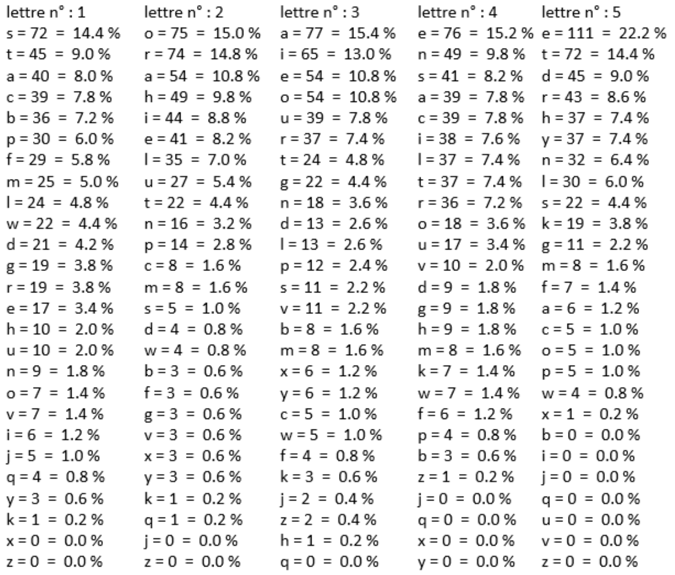
Nous avons commencé par chercher des mots que nous connaissions en anglais et de les classer dans un tableur. Après cela nous avons convenus que nous n'avions pas assez de mots. Alors nous avons cherché des listes sur des dictionnaires et site sur internet. Ensuite nous avons voulu classer également ces listes de mots dans un tableur, mais cela prenez trop de temps et n'était pas très utile. Nous avons donc décidé de faire un programme.

10 Le programme

Pour créer notre programme, nous avons cherché une formule théorique qui nous donnerait le nombre de combinaison possible pour des mots à X lettres. La formule étant

$$\binom{n}{26} \cdot n^m$$

n étant le nombre de lettre parmi 26 sans réutilisation de la même lettre pour un même barillet. m étant le nombre de barillet. Le programme final consistait à entrer une liste de mots commun de la langue anglaise dans un fichier. Dans un notre fichier, nous rentrions les lettes qui nous intéresser d'avoir dans les mots et le programme nous donner une liste de mots avec les lettres choisies. L'intérêt de ce programme étant de trouver des mots avec un maximum de lettres communes et dans un même emplacement (même place dans le mot). Avec ce programme, un notre groupe de notre lycée a fait des probabilités sur l'apparition des lettres dans chacun des barillets.



lettre n° : 1	lettre n° : 2	lettre n° : 3	lettre n° : 4	lettre n° : 5
s = 72 = 14.4 %	o = 75 = 15.0 %	a = 77 = 15.4 %	e = 76 = 15.2 %	e = 111 = 22.2 %
t = 45 = 9.0 %	r = 74 = 14.8 %	i = 65 = 13.0 %	n = 49 = 9.8 %	t = 72 = 14.4 %
a = 40 = 8.0 %	a = 54 = 10.8 %	e = 54 = 10.8 %	s = 41 = 8.2 %	d = 45 = 9.0 %
c = 39 = 7.8 %	h = 49 = 9.8 %	o = 54 = 10.8 %	a = 39 = 7.8 %	r = 43 = 8.6 %
b = 36 = 7.2 %	i = 44 = 8.8 %	u = 39 = 7.8 %	c = 39 = 7.8 %	h = 37 = 7.4 %
p = 30 = 6.0 %	e = 41 = 8.2 %	r = 37 = 7.4 %	i = 38 = 7.6 %	y = 37 = 7.4 %
f = 29 = 5.8 %	l = 35 = 7.0 %	t = 24 = 4.8 %	l = 37 = 7.4 %	n = 32 = 6.4 %
m = 25 = 5.0 %	u = 27 = 5.4 %	g = 22 = 4.4 %	t = 37 = 7.4 %	l = 30 = 6.0 %
l = 24 = 4.8 %	t = 22 = 4.4 %	n = 18 = 3.6 %	r = 36 = 7.2 %	s = 22 = 4.4 %
w = 22 = 4.4 %	n = 16 = 3.2 %	d = 13 = 2.6 %	o = 18 = 3.6 %	k = 19 = 3.8 %
d = 21 = 4.2 %	p = 14 = 2.8 %	l = 13 = 2.6 %	u = 17 = 3.4 %	g = 11 = 2.2 %
g = 19 = 3.8 %	c = 8 = 1.6 %	p = 12 = 2.4 %	v = 10 = 2.0 %	m = 8 = 1.6 %
r = 19 = 3.8 %	m = 8 = 1.6 %	s = 11 = 2.2 %	d = 9 = 1.8 %	f = 7 = 1.4 %
e = 17 = 3.4 %	s = 5 = 1.0 %	v = 11 = 2.2 %	g = 9 = 1.8 %	a = 6 = 1.2 %
h = 10 = 2.0 %	d = 4 = 0.8 %	b = 8 = 1.6 %	h = 9 = 1.8 %	c = 5 = 1.0 %
u = 10 = 2.0 %	w = 4 = 0.8 %	m = 8 = 1.6 %	m = 8 = 1.6 %	o = 5 = 1.0 %
n = 9 = 1.8 %	b = 3 = 0.6 %	x = 6 = 1.2 %	k = 7 = 1.4 %	p = 5 = 1.0 %
o = 7 = 1.4 %	f = 3 = 0.6 %	y = 6 = 1.2 %	w = 7 = 1.4 %	w = 4 = 0.8 %
v = 7 = 1.4 %	g = 3 = 0.6 %	c = 5 = 1.0 %	f = 6 = 1.2 %	x = 1 = 0.2 %
i = 6 = 1.2 %	v = 3 = 0.6 %	w = 5 = 1.0 %	p = 4 = 0.8 %	b = 0 = 0.0 %
j = 5 = 1.0 %	x = 3 = 0.6 %	f = 4 = 0.8 %	b = 3 = 0.6 %	i = 0 = 0.0 %
q = 4 = 0.8 %	y = 3 = 0.6 %	k = 3 = 0.6 %	z = 1 = 0.2 %	j = 0 = 0.0 %
y = 3 = 0.6 %	k = 1 = 0.2 %	j = 2 = 0.4 %	j = 0 = 0.0 %	q = 0 = 0.0 %
k = 1 = 0.2 %	q = 1 = 0.2 %	z = 2 = 0.4 %	q = 0 = 0.0 %	u = 0 = 0.0 %
x = 0 = 0.0 %	j = 0 = 0.0 %	h = 1 = 0.2 %	x = 0 = 0.0 %	v = 0 = 0.0 %
z = 0 = 0.0 %	z = 0 = 0.0 %	q = 0 = 0.0 %	y = 0 = 0.0 %	z = 0 = 0.0 %

Figure 9: Probabilité d'apparition des lettres en fonction de la place occupée dans le mot

11 Conclusion

Utilizing notions of informatics, combinatorics and linguistics has enabled the creation of a code model for the padlock that would guarantee the security desired by the user. The projected model is both easy to implement and user-friendly. It is easily adaptable according to the level of security desired by the user (this is achieved by the extension of the number of letters). The level of security is a very good one thanks to the diminution of the risk represented by the implementation of a common and frequently used password, the latter being replaced with words that are only familiar to the user, while at the same time maintaining a sufficient number of possible variants that would provide security of use.

Notes

[1] Colegiul Național “Emil Racoviță”
[2] Lycée Climatique D’Altitude de Briançon