



# FORRITUNARKEPPNI FRAMHALDSSKÓLANNA

2019

Lausnir á völdum dæmum

## Dæmahöfundar

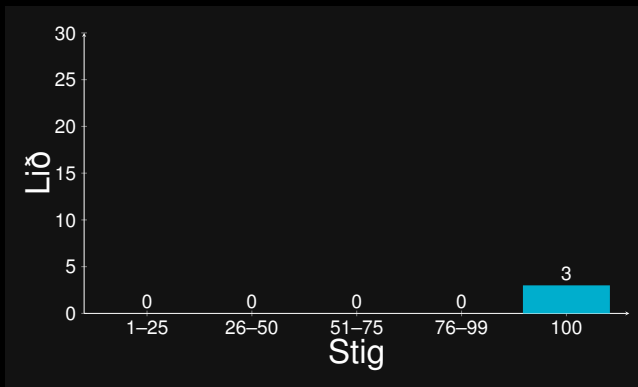
- Arnar Bjarni Arnarson
- Bernhard Linn Hilmarsson
- Bjarki Ágúst Guðmundsson
- Hannes Kristján Hannesson
- Sigurður Jens Albertsson
- Unnar Freyr Erlendsson

## Dæmayfirferð

- Arnar Páll Jóhannsson
- Atli Fannar Franklín
- Bergur Snorrason
- Gísli Freyr
- Guðni Nathan Gunnarsson
- Jón Ingi Ólafsson
- Kieran Horgan
- Ludo Pulles
- Ómar Högni Guðmarsson

# Klósettröð

	Keppendur	Dómarar
Stysta lausn	12	4
Lengsta lausn	18	31
	Tími	Lið
Fyrsta lausn	3:49:05	Bjarni Dagur Thor Kárason



# Klósettröð

## Dæmið

Keppendur þurfa að fara á klósettið, en þeir eru mismikið í speng. Skriðu út númer keppendanna í lækkandi röð eftir því hversu mikið þeir þurfa að fara á klósettið.

## Sýnidæmi

### ■ Inntak:

3

1 2 3

### ■ Úttak:

3 2 1

## Sýnidæmi

### ■ Inntak:

5

1 3 2 5 4

### ■ Úttak:

4 5 2 3 1

## Stærra sýnidæmi

Hve mikið í spreng: 2 3 8 6 7 1 5 4

## Stærra sýnidæmi

Númer keppanda:	1	2	3	4	5	6	7	8
Hve mikið í spreng:	2	3	8	6	7	1	5	4

## Stærra sýnidæmi

Númer keppanda:	1	2	3	4	5	6	7	8
Hve mikið í spreng:	2	3	8	6	7	1	5	4



## Stærra sýnidæmi

Númer keppanda:	1	2	3	4	5	6	7	8
Hve mikið í spreng:	2	3	8	6	7	1	5	4

## Stærra sýnidæmi

Númer keppanda:	1	2	4	5	6	7	8
Hve mikið í spreng:	2	3	6	7	1	5	4

■ Úttak: 3

## Stærra sýnidæmi

Númer keppanda:	1	2	4	5	6	7	8
Hve mikið í spreng:	2	3	6	7	1	5	4

■ Úttak: 3

## Stærra sýnidæmi

Númer keppanda:	1	2	4	5	6	7	8
Hve mikið í spreng:	2	3	6	7	1	5	4

■ Úttak: 3

## Stærra sýnidæmi

Númer keppanda:	1	2	4	6	7	8
Hve mikið í spreng:	2	3	6	1	5	4

- Úttak: 3, 5

## Stærra sýnidæmi

- Úttak: 3, 5, 4, 7, 8, 2, 1, 6

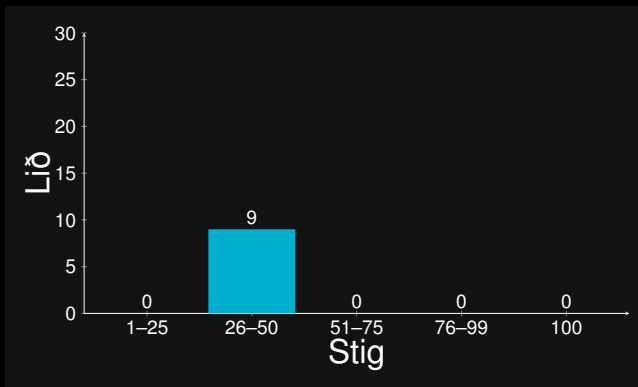
# Klósettröð

## Kóði

```
1 n = int(input())
2 spreng = [ int(s) for s in input().split() ]
3 buinn = [ False for i in range(n) ]
4
5 for i in range(n):
6     mest = 0
7     mestKeppandi = None
8     for j in range(n):
9         if not buinn[j] and spreng[j] > mest:
10             mest = spreng[j]
11             mestKeppandi = j
12
13     print(mestKeppandi + 1)
14     buinn[mestKeppandi] = True
```

# Stalínröðun

	Keppendur	Dómararar
Stysta lausn	?	36
Lengsta lausn	?	45
<hr/>		
	Tími	Lið
Fyrsta lausn	?	?





## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

1 7 5 8 6 3 2 4

## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

1 7 5 8 6 3 2 4

## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

1 7 5 8 6 3 2 4

## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

1 7 5 8 6 3 2 4

## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

1 7 5 8 6 3 2 4  
5 6 3 2 4

## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

1 7 5 8 6 3 2 4  
5 6 3 2 4

## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

1 7 5 8 6 3 2 4  
5 6 3 2 4



## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

```
1 7 5 8 6 3 2 4
5 6 3 2 4
3 2 4
```

## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

```
1 7 5 8 6 3 2 4
5 6 3 2 4
3 2 4
```

## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

```
1 7 5 8 6 3 2 4
5 6 3 2 4
3 2 4
```

## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

```
1 7 5 8 6 3 2 4
5 6 3 2 4
3 2 4
2
```

## Dæmið

Gefinn listi af tölum finndu fjölda ítrana sem það tekur að fjarlægja öll stök sem eru í vaxandi röð þangað til við endum með tóman lista.

## Sýnidæmi

```
1 7 5 8 6 3 2 4
5 6 3 2 4
3 2 4
2
```

## Lausnarhugmynd

Getum hermt eftir ferlinu og ítrekað labbað í gegnum listann og fjarlægt stök ef þau eru allavega jafn stór og öll stök sem koma á undan.

## Lausnarhugmynd

Getum hermt eftir ferlinu og ítrekað labbað í gegnum listann og fjarlægt stök ef þau eru allavega jafn stór og öll stök sem koma á undan.

Tímaflækjan á þessu er  $O(n^2)$ .

## Lausnarhugmynd

Getum hermt eftir ferlinu og ítrekað labbað í gegnum listann og fjarlægt stök ef þau eru allavega jafn stór og öll stök sem koma á undan.

Tímaflækjan á þessu er  $O(n^2)$ . Getum við hermt eftir þessu ferli á hraðari máta?



## Lausn frh.

Þegar við komum að staki  $i$  í listanum viljum við í hvaða ítrun það mun detta út.

## Lausn frh.

Þegar við komum að staki  $i$  í listanum viljum við í hvaða ítrun það mun detta út.

Búum til nýjan lista  $L$  sem byrjar tómur

## Lausn frh.

Þegar við komum að staki  $i$  í listanum viljum við í hvaða ítrun það mun detta út.

Búum til nýjan lista  $L$  sem byrjar tómur

Þegar við komum að staki  $i$  þá löbbum við í gegnum öll stök  $x$  í  $L$  og höldum utan um stærsta gildið  $x$  þannig að  $x \geq a_i$

## Lausn frh.

Þegar við komum að staki  $i$  í listanum viljum við í hvaða ítrun það mun detta út.

Búum til nýjan lista  $L$  sem byrjar tómur

Þegar við komum að staki  $i$  þá löbbum við í gegnum öll stök  $x$  í  $L$  og höldum utan um stærsta gildið  $x$  þannig að  $x \geq a_i$

Breytum  $x$  í  $a_i$ , ef það finnst ekkert gildi  $x \leq a_i$  þá bætum við  $a_i$  aftast í  $L$

## Lausn frh.

Þegar við komum að staki  $i$  í listanum viljum við í hvaða ítrun það mun detta út.

Búum til nýjan lista  $L$  sem byrjar tómur

Þegar við komum að staki  $i$  þá löbbum við í gegnum öll stök  $x$  í  $L$  og höldum utan um stærsta gildið  $x$  þannig að  $x \geq a_i$

Breytum  $x$  í  $a_i$ , ef það finnst ekkert gildi  $x \leq a_i$  þá bætum við  $a_i$  aftast í  $L$

Svarið er núna fjöldi staka í  $L$

## Lausn frh.

Þegar við komum að staki  $i$  í listanum viljum við í hvaða ítrun það mun detta út.

Búum til nýjan lista  $L$  sem byrjar tómur

Þegar við komum að staki  $i$  þá löbbum við í gegnum öll stök  $x$  í  $L$  og höldum utan um stærsta gildið  $x$  þannig að  $x \geq a_i$

Breytum  $x$  í  $a_i$ , ef það finnst ekkert gildi  $x \leq a_i$  þá bætum við  $a_i$  aftast í  $L$

Svarið er núna fjöldi staka í  $L$

Tímaflækjan á þessu er ennþá  $O(n^2)$  en við getum gert betur.

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ \quad ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ \quad ]$



## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 1 \quad ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 1 \quad ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 7 \quad ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 7 \quad ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 7 \ 5 \quad ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 7 \ 5 \ ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 8 \ 5 \ ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 8 \ 5 \ ]$



## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 8 \ 6 \ ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 8 \ 6 \ ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 8 \ 6 \ 3 \ ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 8 \ 6 \ 3 \ ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 8 \ 6 \ 3 \ 2 ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 8 \ 6 \ 3 \ 2 ]$

## Sýnidæmi

1 7 5 8 6 3 2 4

$L = [ 8 \ 6 \ 4 \ 2 ]$

Lausn frh.

Getum tekið eftir að  $L$  mun alltaf vera raðaður



## Lausn frh.

Getum tekið eftir að  $L$  mun alltaf vera raðaður

Í stað þess að skoða línulega öll stök í  $L$  þá getum við beitt helmingunarleit á  $L$  og getum þá fundið réttan stað fyrir  $a_i$  í  $O(\log(k))$

## Lausn frh.

Getum tekið eftir að  $L$  mun alltaf vera raðaður

Í stað þess að skoða línulega öll stök í  $L$  þá getum við beitt helmingunarleit á  $L$  og getum þá fundið réttan stað fyrir  $a_i$  í  $O(\log(k))$

Heildartímaflækjan er núna  $O(n \cdot \log(k))$  þar sem svarið er  $k$

## Lausn 2

Fjöldi vaxandi raða í lista er jafnt lengd lengstu minnkandi hlutrunu í lista.

## Lausn 2

Fjöldi vaxandi raða í lista er jafnt lengd lengstu minnkandi hlutrunu í lista.

Að finna lengstu minnkandi hlutrunu í lista er jafngilt því að finna lengstu vaxandi hlutrunu ef þú snýrð við listanum (eða margfaldar öll gildi með  $-1$ )

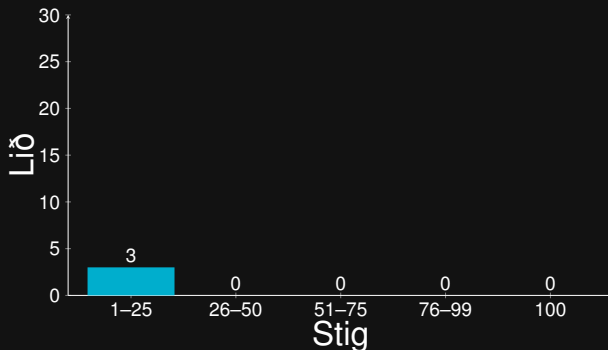
## Lausn 2

Fjöldi vaxandi raða í lista er jafnt lengd lengstu minnkandi hlutrunu í lista.

Að finna lengstu minnkandi hlutrunu í lista er jafngilt því að finna lengstu vaxandi hlutrunu ef þú snýrð við listanum (eða margfaldar öll gildi með  $-1$ )

Getum fundið lengstu hlutrunu í tíma  $O(n \cdot \log(n))$  og skrifað síðan út lengdina á henni

	Keppendur	Dómararar
Stysta lausn	?	48
Lengsta lausn	?	94
<hr/>		
	Tími	Lið
Fyrsta lausn	?	?



## Dæmið

Gefið er lýsinga á hvernig strengur er skrifaður. Seigja fyrir alla mögulega strengi sem gæti verið skrifa hvert mesta magn strokleðurs myndi þurfa í versta falli.

## Dæmið

Gefið er lýsinga á hvernig strengur er skrifaður. Seigja fyrir alla mögulega strengi sem gæti verið skrifa hvert mesta magn strokleðurs myndi þurfa í versta falli.

## Lausn

- Telja fyrir hverja staðsetningu hversu oft stafurinn þar er strokaður út.
- Setja staf með hæsta kostnað við að stroka út á staðsetningu sem er oftast strokað út.



## Dæmið

Gefið er lýsinga á hvernig strengur er skrifaður. Seigja fyrir alla mögulega strengi sem gæti verið skrifa hvert mesta magn strokleðurs myndi þurfa í versta falli.

## Lausn

- Telja fyrir hverja staðsetningu hversu oft stafurinn þar er strokaður út.
- Setja staf með hæsta kostnað við að stroka út á staðsetningu sem er oftast strokað út.

Hvernig gerum við þetta hratt?

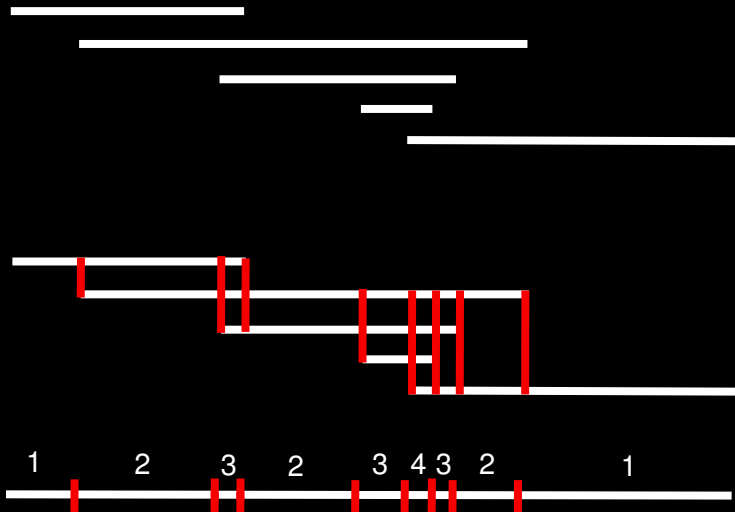
## Lausn

- Skilgreinum í hver skifti sem er strokað út  $y_i$  stafi frá staðsetningu  $at$  með bili  $(at - y_i, at)$

## Lausn

- Skilgreinum í hver skifti sem er strokað út  $y_i$  stafi frá staðsetningu  $at$  með bili  $(at - y_i, at)$
- Röðum svo öllum bilum eftir byrjunarpunkt og svo eftir endapunkt

# Skrift



## Lausn

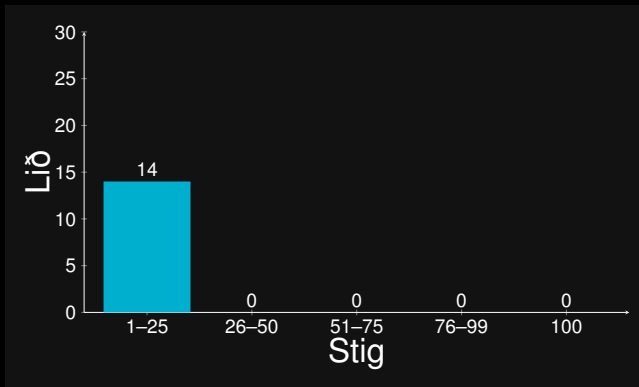
- Það munu vera mestalagi  $2 \cdot q$  bil og getum nú parað þessum bilum gráðugt við stafi sem hafa hæsta kostnað við að stroka út.

## Lausn

- Það munu vera mestalagi  $2 \cdot q$  bil og getum nú parað þessum bilum gráðugt við stafi sem hafa hæsta kostnað við að stroka út.
- Heildar tímaflækja  $O(m \cdot \log m + q \cdot \log q)$

# Gagnaleki

	Keppendur	Dómararar
Stysta lausn	?	64
Lengsta lausn	?	2500
	Tími	Lið
Fyrsta lausn	?	?



## Dæmið

Forritun.is var hakkað og 2500 tættum lykilorðum keppenda lekið á netið. Brjóttu eins mörg þeirra og þú getur.

## Lausn 1 - “Brute force”

- Byrja á að búa til lykilorð sem koma til greina, t.d.:
  - Tölur minni en 10 000
  - Ensk orð — hægt að finna orðalista á netinu
  - Ensk orð með tölum fyrir aftan
- Prófa að tæta hvert einasta lykilorð
- Getum gert þetta á okkar eigin tölum



## Lausn 1 - “Brute force”

```
1 for taettLykilord in gagnaleki:
2     for lykilord in mogulegLykilord:
3         if taettLykilord == taetaLykilord(lykilord):
4             print('%s:%s' % (taettLykilord, lykilord))
```

- Það eru 2500 tætt lykilorð
- Segjum að við ætlum að prófa  $n$  lykilorð, og það tekur  $x$  sekúndur að tæta lykilorð
- Þessi keyrsla mun taka  $2500 \cdot n \cdot x$  sekúndur að keyra

## Lausn 1 - “Brute force”

```
1 for taettLykilord in gagnaleki:
2     for lykilord in mogulegLykilord:
3         if taettLykilord == taetaLykilord(lykilord):
4             print('%s:%s' % (taettLykilord, lykilord))
```

## Lausn 1 - “Brute force”

```
1 for lykilord in mogulegLykilord:
2     for taettLykilord in gagnaleki:
3         if taettLykilord == taetaLykilord(lykilord):
4             print('%s:%s' % (taettLykilord, lykilord))
```

## Lausn 1 - "Brute force"

```
1 for lykilord in mogulegLykilord:
2     taettLykilord2 = taetaLykilord(lykilord)
3     for taettLykilord in gagnaleki:
4         if taettLykilord == taettLykilord2:
5             print('%s:%s' % (taettLykilord, lykilord))
```

## Lausn 1 - “Brute force”

```
1 gagnaleki = set(gagnaleki)
2 for lykilord in mogulegLykilord:
3     taettLykilord2 = taetaLykilord(lykilord)
4     if taettLykilord2 in gagnaleki:
5         print('%s:%s' % (taettLykilord2, lykilord))
```

- Núna mun keyrslan bara taka  $n \cdot x$  sekúndur að keyra
- Getum prófað 2500 sinnum fleiri lykilorð á sama tíma
- Tíminn sem það tekur að tæta lykilorð,  $x$ , er ennþá hár

## Lausn 1 - “Brute force”

```
1 def brengla(s, at, u):
2     magic = 'b058592efd277ae75f27bd99d1628fbd'
3     if at >= len(s):
4         return magic
5
6     res = leggjaSaman(brengla(s, at+1, True),
7                       brengla(s, at+1, False))
8     for i in range(6):
9         res = leggjaSaman(res, res)
10
11    cnt = ord(s[at])
12    for i in range(cnt):
13        res = leggjaSaman(res, magic)
14
15    return res
16
17 def taetaLykilord(s):
18    return brengla(s, 0, True)
```

## Lausn 1 - "Brute force"

```
1 def brengla(s, at):
2     magic = 'b058592efd277ae75f27bd99d1628fbd'
3     if at >= len(s):
4         return magic
5
6     res = leggjaSaman(brengla(s, at+1),
7                       brengla(s, at+1))
8     for i in range(6):
9         res = leggjaSaman(res, res)
10
11    cnt = ord(s[at])
12    for i in range(cnt):
13        res = leggjaSaman(res, magic)
14
15    return res
16
17 def taetaLykilord(s):
18    return brengla(s, 0)
```

## Lausn 1 - “Brute force”

```
1 def brengla(s, at):
2     magic = 'b058592efd277ae75f27bd99d1628fbd'
3     if at >= len(s):
4         return magic
5
6     brengl = brengla(s, at+1)
7     res = leggjaSaman(brengl,
8                       brengl)
9     for i in range(6):
10        res = leggjaSaman(res, res)
11
12    cnt = ord(s[at])
13    for i in range(cnt):
14        res = leggjaSaman(res, magic)
15
16    return res
17
18 def taetaLykilord(s):
19    return brengla(s, 0)
```



## Lausn 1 - "Brute force"

```
1 def brengla(s, at):
2     magic = 'b058592efd277ae75f27bd99d1628fbd'
3     if at >= len(s):
4         return magic
5
6     res = brengla(s, at+1)
7     for i in range(7):
8         res = leggjaSaman(res, res)
9
10    cnt = ord(s[at])
11    for i in range(cnt):
12        res = leggjaSaman(res, magic)
13
14    return res
15
16 def taetaLykilord(s):
17    return brengla(s, 0)
```

## Lausn 1 - “Brute force”

```
1 def brengla(s, at):
2     magic = 'b058592efd277ae75f27bd99d1628fbd'
3     if at >= len(s):
4         return magic
5
6     res = brengla(s, at+1)
7     for i in range(7):
8         res = leggjaSaman(res, res)
9
10    cnt = ord(s[at])
11    for i in range(cnt):
12        res = leggjaSaman(res, magic)
13
14    return res
15
16 def taetaLykilord(s):
17    return brengla(s, 0)
```

- Tímaflækjan var  $O(2^\ell)$ , en núna  $O(\ell)$  fyrir  $\ell$  stafa streng

## Lausn 2 - Finna mynstur

```
1  ed15a97d084bf9f8ca7a1759c52561ad
2  9d6e02ac057374e029a1d4f39687f16a
3  4dc65bdb029aefc788c9928d67ea8127
4  fe1eb509ffc26aaee7f15027394d10e4
5  ae770e38fce9e59647190dc10aafa0a1
6  5ecf6767fa11607da640cb5adc12305e
7  0f27c096f738db65056888f4ad74c01b
8  bf8019c5f460564c6490468e7ed74fd8
9  6fd872f4f187d133c3b804285039df95
```

## Lausn 2 - Finna mynstur

```
11 4bbdd0829a8b82ae73a6f753a68e59ad
12 77ea68012e48f65e0785c43c57d6382d
13 a416ff7fc2066a0d9b649125091e16ad
14 d04396fe55c3ddb2f435e0dba65f52d
15 fc702e7ce981516cc3222af66badd3ad
16 289cc5fb7d3ec51c5700f7df1cf5b22d
17 54c95d7a10fc38cbeadfc4c7ce3d90ad
18 80f5f4f8a4b9ac7b7ebe91b07f856f2d
19 ad228c773877202b129d5e9930cd4dad
20 cfe9923303f589e63eefe804c6a90aea
21 fc1629b197b2fd95d2ceb4ed77f0e96a
22 2842c1302b70714566ad81d62938c7ea
23 546f58aebf2de4f4fa8c4ebeda80a66a
```

## Lausn 2 - Finna mynstur

```
11 4bbdd0829a8b82ae73a6f753a68e59ad
12 77ea68012e48f65e0785c43c57d6382d
13 a416ff7fc2066a0d9b649125091e16ad
14 d04396fe55c3ddb2f435e0dba65f52d
15 fc702e7ce981516cc3222af66badd3ad
16 289cc5fb7d3ec51c5700f7df1cf5b22d
17 54c95d7a10fc38cbeadfc4c7ce3d90ad
18 80f5f4f8a4b9ac7b7ebe91b07f856f2d
19 ad228c773877202b129d5e9930cd4dad
20 cfe9923303f589e63eefe804c6a90aea
21 fc1629b197b2fd95d2ceb4ed77f0e96a
22 2842c1302b70714566ad81d62938c7ea
23 546f58aebf2de4f4fa8c4ebeda80a66a
```

## Lausn 2 - Finna mynstur

```
11 4bbdd0829a8b82ae73a6f753a68e59ad
12 77ea68012e48f65e0785c43c57d6382d
13 a416ff7fc2066a0d9b649125091e16ad
14 d04396fe55c3ddb2f435e0dba65f52d
15 fc702e7ce981516cc3222af66badd3ad
16 289cc5fb7d3ec51c5700f7df1cf5b22d
17 54c95d7a10fc38cbeadfc4c7ce3d90ad
18 80f5f4f8a4b9ac7b7ebe91b07f856f2d
19 ad228c773877202b129d5e9930cd4dad
20 cfe9923303f589e63eefe804c6a90aea
21 fc1629b197b2fd95d2ceb4ed77f0e96a
22 2842c1302b70714566ad81d62938c7ea
23 546f58aebf2de4f4fa8c4ebeda80a66a
```

## Lausn 2 - Finna mynstur

```
111 9fd1534bba4fdd830a16f4445b0a59ad
112 b61d12959909b54cf97d689cfef999ad
113 cc68d1df77c38d16e8e3dcf5a2e8d9ad
114 e2b49129567d64e0d84a514e46d819ad
115 f900507335373caac7b0c5a6eac759ad
116 0f4c0fbd13f11474b71739ff8eb699ad
117 2597cf06f2aaec3ea67dae5832a5d9ad
118 3be38e50d164c40895e422b0d69519ad
119 522f4d9ab01e9bd2854a97097a8459ad
120 b5b22b806f537968ae8f4cd46862f82d
121 cbfdeaca4e0d51329df5c12d0c52382d
122 e249aa142cc728fc8d5c3585b041782d
123 f895695e0b8100c67cc2a9de5430b82d
```

## Lausn 2 - Finna mynstur

```
111 9fd1534bba4fdd830a16f4445b0a59ad
112 b61d12959909b54cf97d689cfef999ad
113 cc68d1df77c38d16e8e3dcf5a2e8d9ad
114 e2b49129567d64e0d84a514e46d819ad
115 f900507335373caac7b0c5a6eac759ad
116 0f4c0fbd13f11474b71739ff8eb699ad
117 2597cf06f2aaec3ea67dae5832a5d9ad
118 3be38e50d164c40895e422b0d69519ad
119 522f4d9ab01e9bd2854a97097a8459ad
120 b5b22b806f537968ae8f4cd46862f82d
121 cbfdeaca4e0d51329df5c12d0c52382d
122 e249aa142cc728fc8d5c3585b041782d
123 f895695e0b8100c67cc2a9de5430b82d
```



## Lausn 2 - Finna mynstur

```
111 9fd1534bba4fdd830a16f4445b0a59ad
112 b61d12959909b54cf97d689cfef999ad
113 cc68d1df77c38d16e8e3dcf5a2e8d9ad
114 e2b49129567d64e0d84a514e46d819ad
115 f900507335373caac7b0c5a6eac759ad
116 0f4c0fbd13f11474b71739ff8eb699ad
117 2597cf06f2aaec3ea67dae5832a5d9ad
118 3be38e50d164c40895e422b0d69519ad
119 522f4d9ab01e9bd2854a97097a8459ad
120 b5b22b806f537968ae8f4cd46862f82d
121 cbfdeaca4e0d51329df5c12d0c52382d
122 e249aa142cc728fc8d5c3585b041782d
123 f895695e0b8100c67cc2a9de5430b82d
```

## Lausn 2 - Finna mynstur

```
1 def einsStafir(a, b):  
2     for l in range(len(a), 0, -1):  
3         if a[-l:] == b[-l:]:  
4             return l  
5     return 0
```

## Lausn 2 - Finna mynstur

```
1 lykilord = ''
2
3 while True:
4     best = 0
5     for d1 in '0123456789':
6         for d2 in '0123456789':
7             taett = taetaLykilord(lykilord + d1 + d2)
8             if taett == taettLykilord:
9                 print('%s:%s' % (taett, lykilord + d1 + d2))
10                sys.exit(0)
11            stafir = einsStafir(taett, taettLykilord)
12            if stafir > best:
13                best = stafir
14                bestiStafur = d1
15
16 lykilord += bestiStafur
```

## Lausn 3 - Skilja tætifallið (vísbending)

```
1 def leggjaSaman(a, b):
2     carry = 0
3     s = ''
4     for at in range(31, -1, -1):
5         carry += heiltala(a[at]) + heiltala(b[at])
6         s = stafur(carry % 16) + s
7         carry = carry // 16
8     return s
9
10 def brengla(s, at):
11     magic = 'b058592efd277ae75f27bd99d1628fbd'
12     if at >= len(s):
13         return magic
14     res = brengla(s, at+1)
15     for i in range(7):
16         res = leggjaSaman(res, res)
17     cnt = ord(s[at])
18     for i in range(cnt):
19         res = leggjaSaman(res, magic)
20     return res
```

## Lausn 3 - Skilja tætifallið (vísbending)

```
1 def leggjaSaman(a, b):
2     return (a+b) % (2**128)
3
4 def brengla(s, at):
5     magic = 0xb058592efd277ae75f27bd99d1628fbd
6     if at >= len(s):
7         return magic
8     res = brengla(s, at+1)
9     for i in range(7):
10        res = leggjaSaman(res, res)
11    cnt = ord(s[at])
12    for i in range(cnt):
13        res = leggjaSaman(res, magic)
14    return res
```

- Minnsti fjöldi lína sem þarf til að leysa öll dæmi í Alfa: 512
- Fjöldi committa í Git repositoryinu okkar: 377
- Heildarfjöldi lína í öllum skráum sem við koma verkefnunum: 9681536