Forritunarkeppni Framhaldsskólanna '16

Lausnir á völdum dæmum

19. mars 2016

Dæmahöfundar

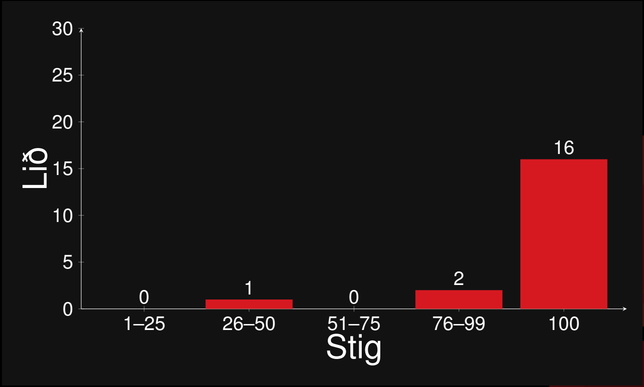
- Bjarki Ágúst Guðmundsson
- Hjalti Magnússon
- Tómas Ken Magnússon

Dæmayfirferð

- Arnar Bjarni Arnarson
- Hlynur Óskar Guðmundsson
- James Elías Sigurðarson
- Kristmundur Ágúst Jónsson
- Unnar Freyr Erlendsson

Giskaðu á svarið

	Keppendur	Dómarar
Stysta lausn	1	1
Lengsta lausn	27	33
	Tími	Lið
Fyrsta lausn	3:34:16	Magic Mike XXL



Giskaðu á svarið

Dæmið

Giska á heiltölu á milli 0 og 1 000.

Lausn

- Búum til forrit sem skrifar út giskið okkar:

```
print 821
```

- Stigafjöldinn segir til um hversu langt við erum frá réttu svari, eða 0 ef við erum lengra en 100 frá

Giskaðu á svarið

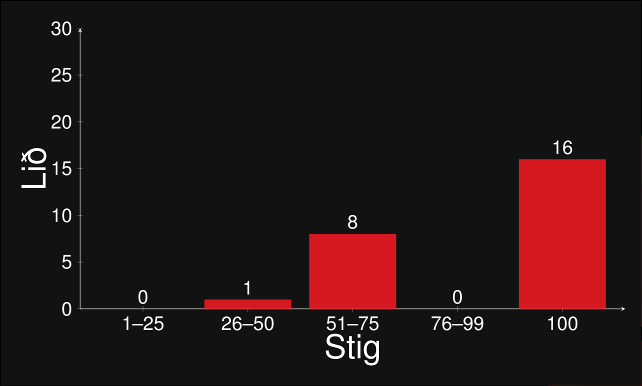
Lausn

- Nóg að giska á 100, 300, 500, 700, 900
- Einhver af þessum tölum mun gefa okkur stigafjölda sem er stærri en 0
- Ef talan X fær stigafjöldann Y (sem er stærri en 0), þá er svarið annaðhvort $X - Y$ eða $X + Y$
- Prufum bæði

- Versta falli 7 gisk

Liðaskipting

	Keppendur	Dómarar
Stysta lausn	5	6
Lengsta lausn	37	27
	Tími	**Lið**
Fyrsta lausn	0:06:59	Los Magos



Liðaskipting

Dæmið

Athuga hvort hægt sé að skipta N keppendum í þriggja manna lið þannig að enginn sé útundan.

Lausn

- Þrír í hverju liði, svo fjöldi liða er $N/3$
- Bara hægt ef deilingin gefur engan afgang

```
print "Jebb" if N % 3 == 0 else "Neibb"
```

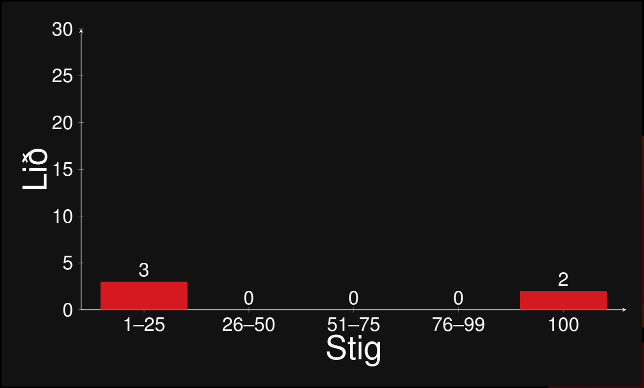
Liðaskipting

Lausn

- $N \leq 10^{100}$
- Nota Python eða BigInteger í C#/Java
- Nota deilireglur:
    - Tala er deilanleg með 3 ef þversumma hennar er deilanleg með 3
    - Bæði 123 og $1 + 2 + 3 = 6$ eru deilanlegar með 3
    - Hvorki 124 né $1 + 2 + 4 = 7$ eru deilanlegar með 3

Röknet

	Keppendur	Dómarar
Stysta lausn	71	18
Lengsta lausn	107	106
	Tími	Lið
Fyrsta lausn	4:38:49	12" Python



Röknet

Dæmið

Gefið röknet og sanngildi sem sett eru inn í netið, hvaða sanngildi koma út?

Lausn

- Ógnvekjanlegt við fyrstu sýn, en...
- *„Ef útgangur hlutar A er tengdur við inngang hlutar B, þá mun hlutur A vera skilgreindur í inntakinu áður en hlutur B er skilgreindur í inntakinu“*
- Göngum í gegnum hlutina í sömu röð og þeir koma í inntakinu
- Ef búið er að reikna sanngildið fyrir alla hlutina á undan, þá getum við reiknað sanngildið fyrir núverandi hlut á einfaldan hátt

Röknet

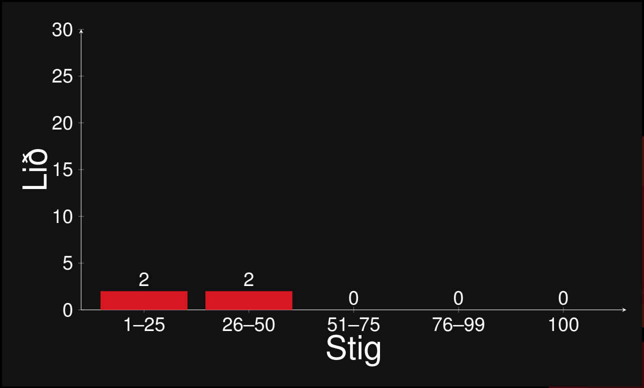
Lausn

- Látum $v[x]$ tákna sanngildi breytunnar x
- Þá verður þetta að einskonar forritunarmáli:

Inntak	Skipun
INNTAK x SATT	$v[x] = \texttt{true}$
INNTAK x OSATT	$v[x] = \texttt{false}$
OG x y z	$v[z] = v[x] \wedge v[y]$
EDA x y z	$v[z] = v[x] \vee v[y]$
EKKI x z	$v[z] = \neg v[x]$
UTTAK x	print $v[x]$

Roomba

	Keppendur	Dómarar
Stysta lausn	?	85
Lengsta lausn	?	118
	Tími	Lið
Fyrsta lausn	?	?



Roomba

Dæmið

Finna stystu rás sem heimsækir alla reiti í $n \times m$ borði.

Lausn

Þrjú tilfelli

- $n = 1$ eða $m = 1$
- Annað hvort n eða m er slétt
- Bæði n og m eru oddatölur

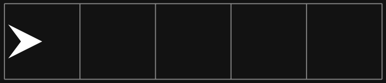
Roomba - Tilfelli 1

$n = 1$ eða $m = 1$

- Eina leiðin er að fara fram og til baka
- Fjöldi hreyfinga er $2(n \cdot m - 1)$

Roomba - Tilfelli 1

$n = 1$, $m = 5$

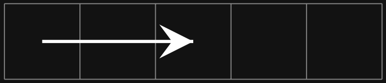


Roomba - Tilfelli 1

$n = 1$, $m = 5$

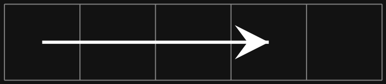
Roomba - Tilfelli 1

$n = 1$, $m = 5$



Roomba - Tilfelli 1

$n = 1$, $m = 5$



Roomba - Tilfelli 1

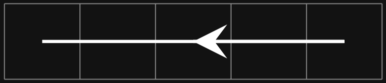
$n = 1$, $m = 5$

Roomba - Tilfelli 1

$n = 1$, $m = 5$

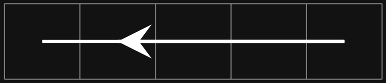
Roomba - Tilfelli 1

$n = 1$, $m = 5$



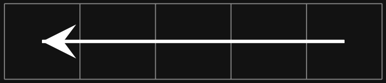
Roomba - Tilfelli 1

$n = 1$, $m = 5$



Roomba - Tilfelli 1

$n = 1$, $m = 5$



Roomba - Tilfelli 2

n eða m slétt

- Alltaf hægt að velja stystu mögulegu leið (Hamilton-rás)
- Fjöldi hreyfinga er $n \cdot m$

Roomba - Tilfelli 2

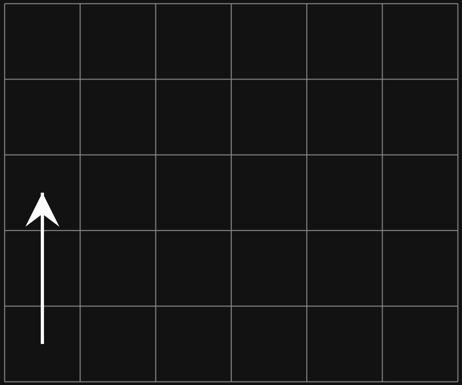
$n = 5$, $m = 6$

Roomba - Tilfelli 2

$n = 5$, $m = 6$

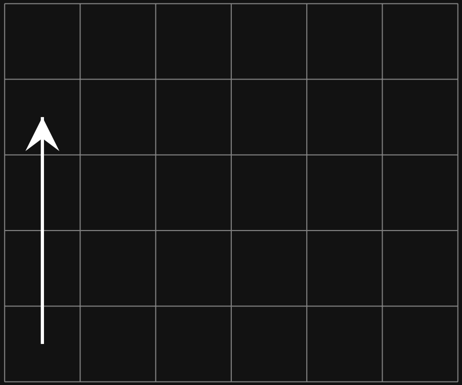
Roomba - Tilfelli 2

$n = 5, m = 6$



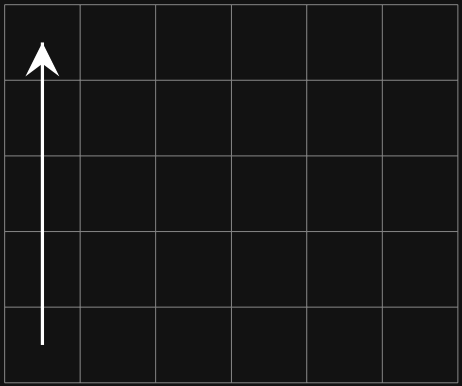
Roomba - Tilfelli 2

$n = 5,\ m = 6$



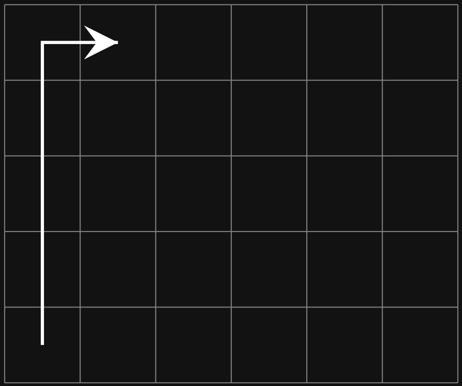
Roomba - Tilfelli 2

$n = 5$, $m = 6$



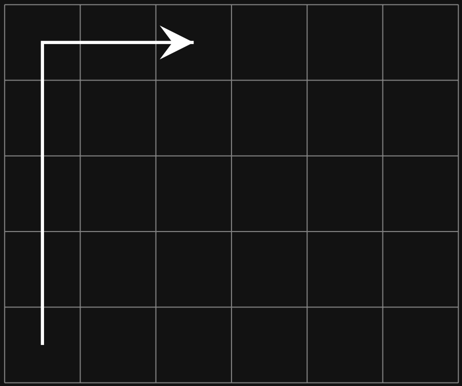
Roomba - Tilfelli 2

$n = 5$, $m = 6$



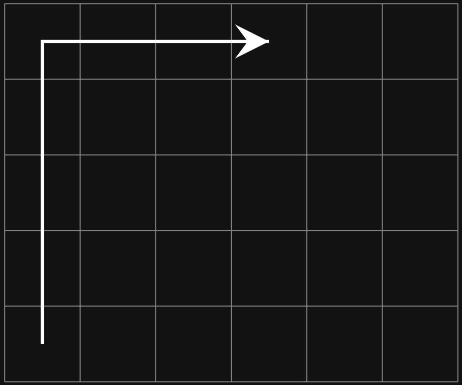
Roomba - Tilfelli 2

$n = 5$, $m = 6$



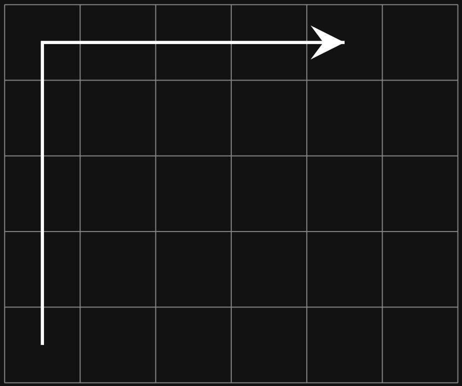
Roomba - Tilfelli 2

$n = 5$, $m = 6$



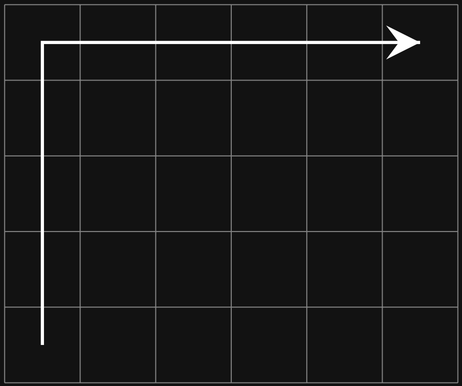
Roomba - Tilfelli 2

$n = 5$, $m = 6$



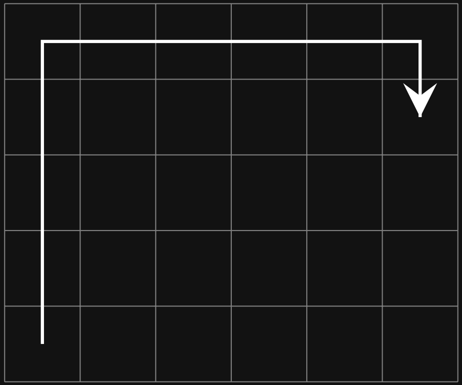
Roomba - Tilfelli 2

$n = 5$, $m = 6$



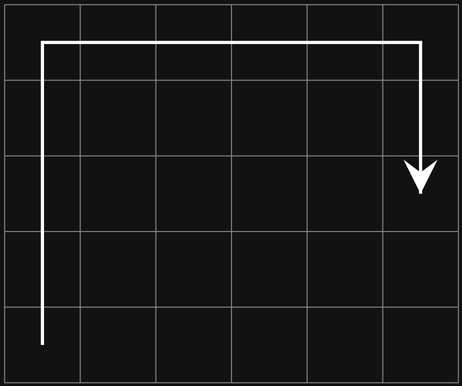
Roomba - Tilfelli 2

$n = 5$, $m = 6$



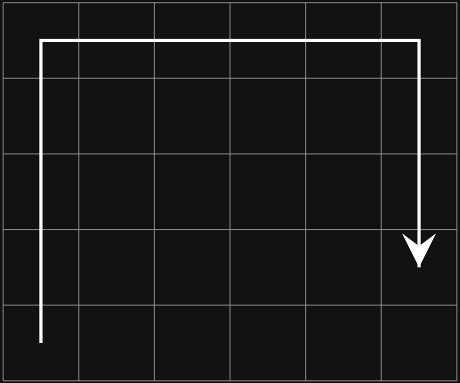
Roomba - Tilfelli 2

$n = 5$, $m = 6$



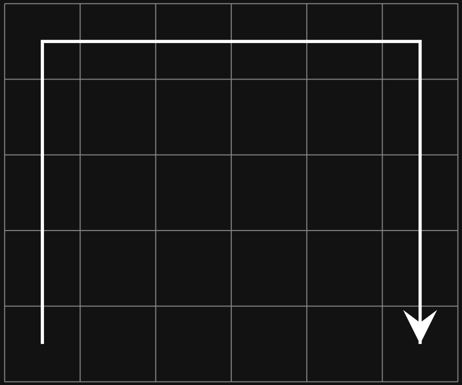
Roomba - Tilfelli 2

$n = 5$, $m = 6$



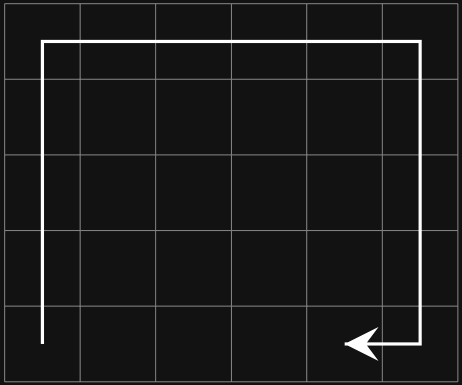
Roomba - Tilfelli 2

$n = 5$, $m = 6$



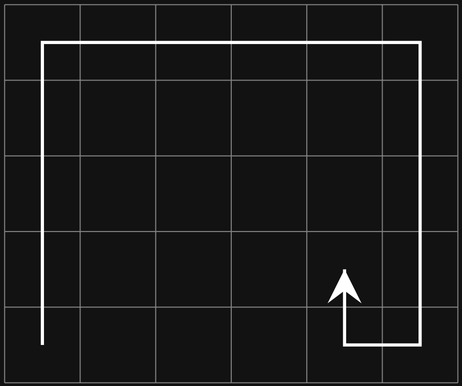
Roomba - Tilfelli 2

$n = 5$, $m = 6$



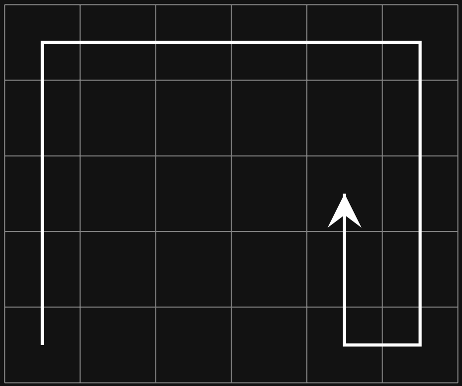
Roomba - Tilfelli 2

$n = 5$, $m = 6$



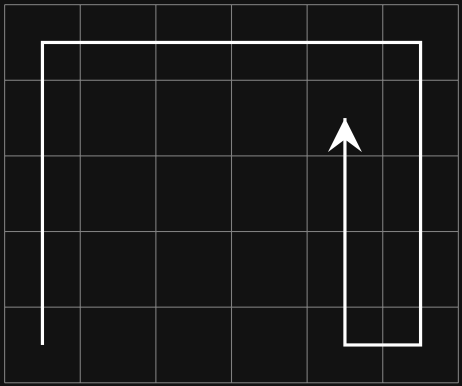
Roomba - Tilfelli 2

$n = 5$, $m = 6$



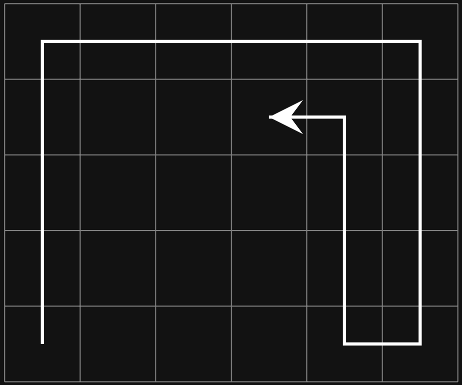
Roomba - Tilfelli 2

$n = 5$, $m = 6$



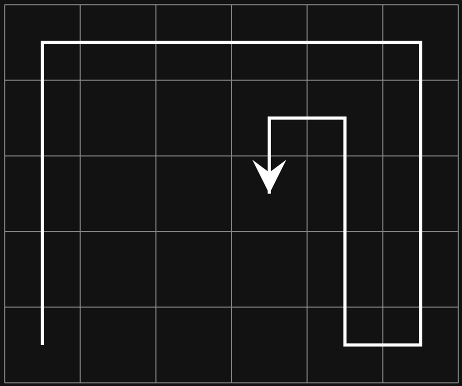
Roomba - Tilfelli 2

$n = 5, m = 6$



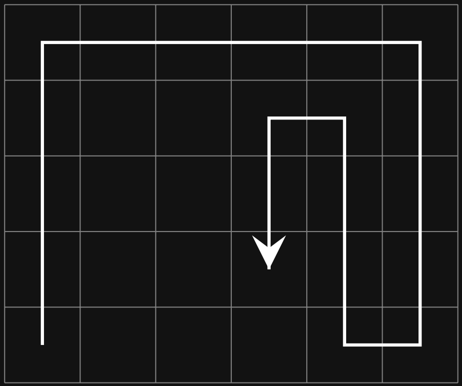
Roomba - Tilfelli 2

$n = 5$, $m = 6$



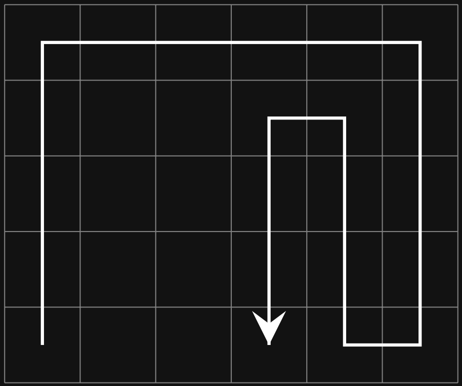
Roomba - Tilfelli 2

$n = 5$, $m = 6$



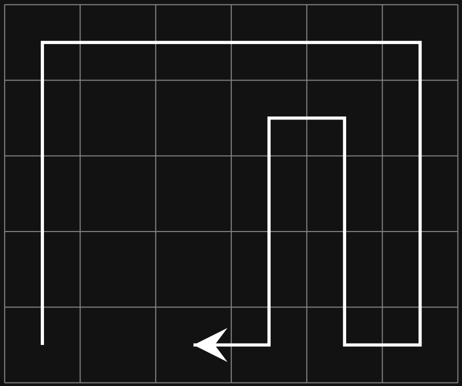
Roomba - Tilfelli 2

$n = 5$, $m = 6$



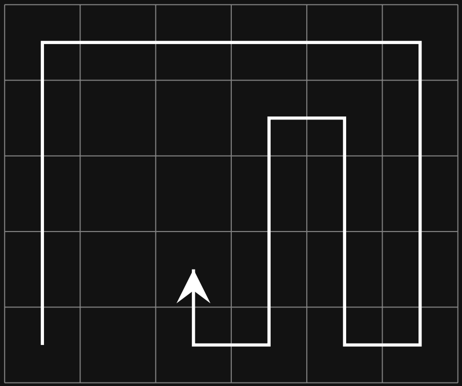
Roomba - Tilfelli 2

$n = 5$, $m = 6$



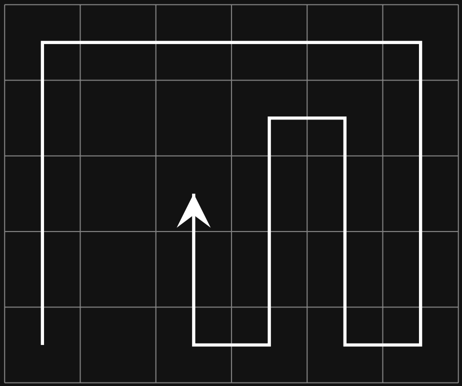
Roomba - Tilfelli 2

$n = 5$, $m = 6$



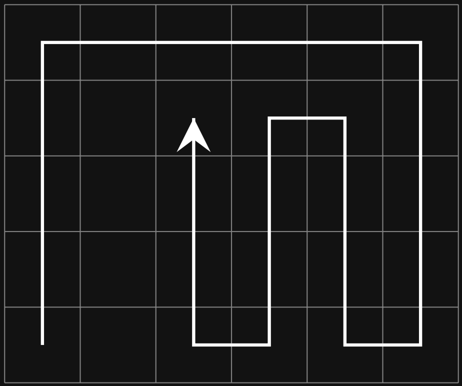
Roomba - Tilfelli 2

$n = 5, m = 6$



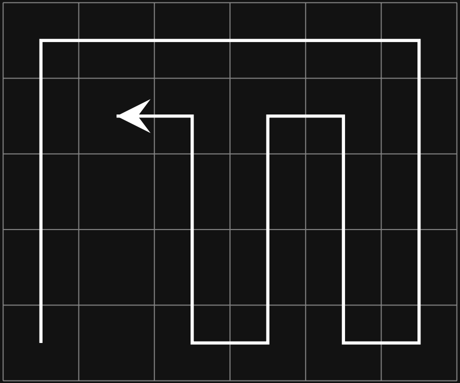
Roomba - Tilfelli 2

$n = 5$, $m = 6$



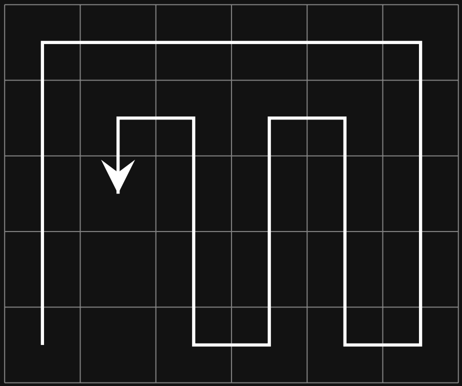
Roomba - Tilfelli 2

$n = 5$, $m = 6$



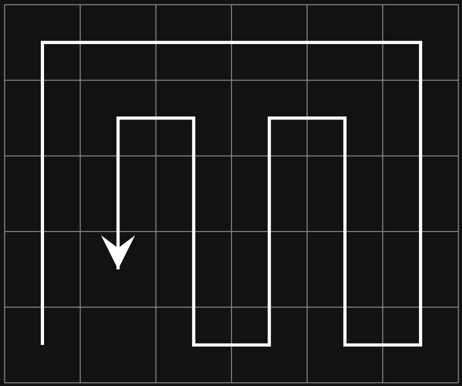
Roomba - Tilfelli 2

$n = 5$, $m = 6$



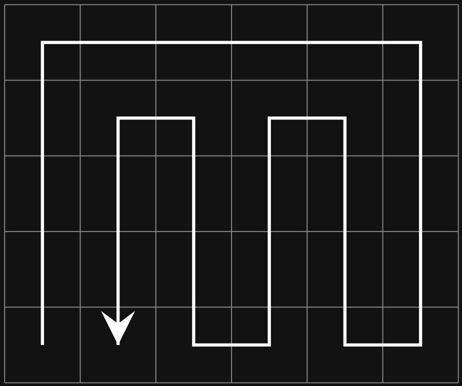
Roomba - Tilfelli 2

$n = 5$, $m = 6$



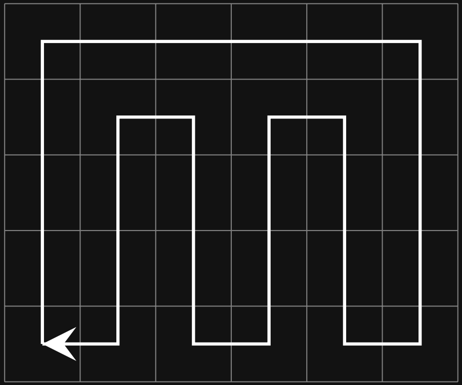
Roomba - Tilfelli 2

$n = 5$, $m = 6$



Roomba - Tilfelli 2

$n = 5$, $m = 6$



Roomba - Tilfelli 3

n og m oddatölur

- Engin leið sem heimsækir hvern reit nákvæmlega einu sinni (engin Hamilton-rás)
- Fjöldi hreyfinga er $n \cdot m + 1$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

Roomba - Tilfelli 3

$n = 5, m = 7$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

Roomba - Tilfelli 3

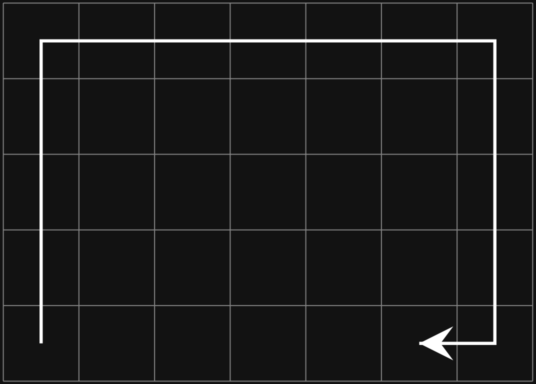
$n = 5$, $m = 7$

Roomba - Tilfelli 3

$n = 5$, $m = 7$

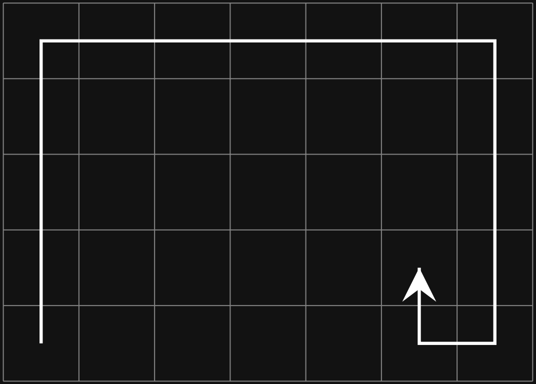
Roomba - Tilfelli 3

$n = 5, m = 7$



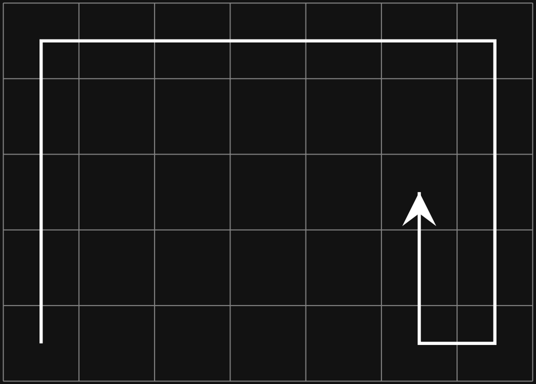
Roomba - Tilfelli 3

$n = 5$, $m = 7$



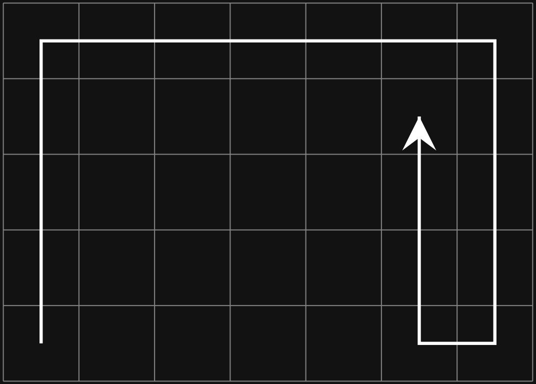
Roomba - Tilfelli 3

$n = 5$, $m = 7$



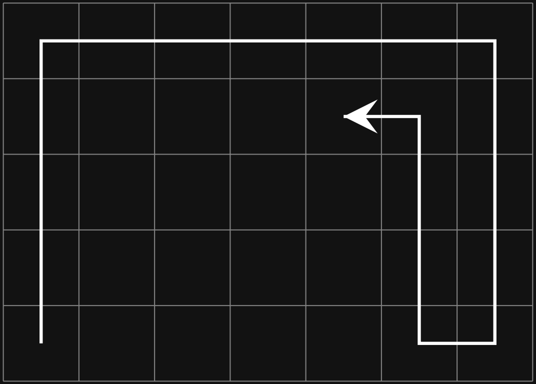
Roomba - Tilfelli 3

$n = 5$, $m = 7$



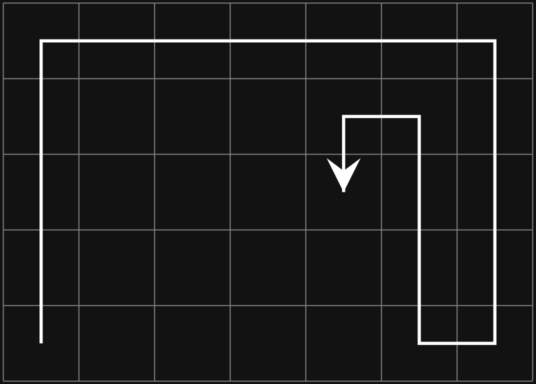
Roomba - Tilfelli 3

$n = 5$, $m = 7$



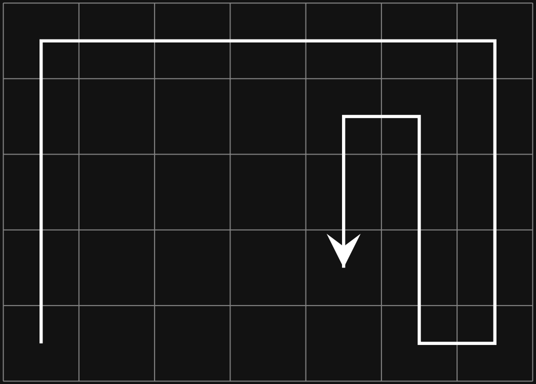
Roomba - Tilfelli 3

$n = 5$, $m = 7$



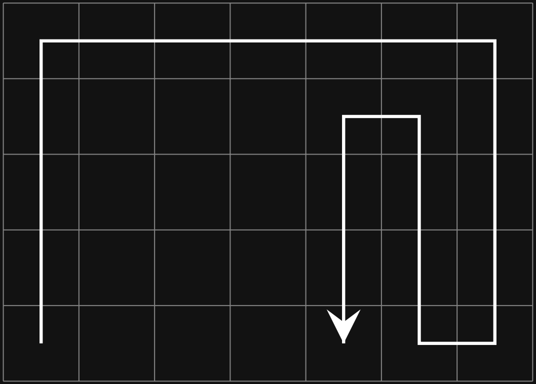
Roomba - Tilfelli 3

$n = 5$, $m = 7$



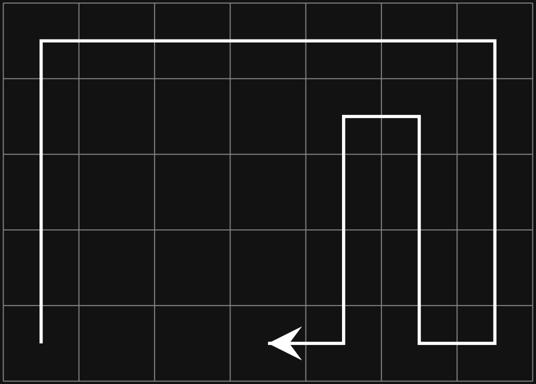
Roomba - Tilfelli 3

$n = 5$, $m = 7$



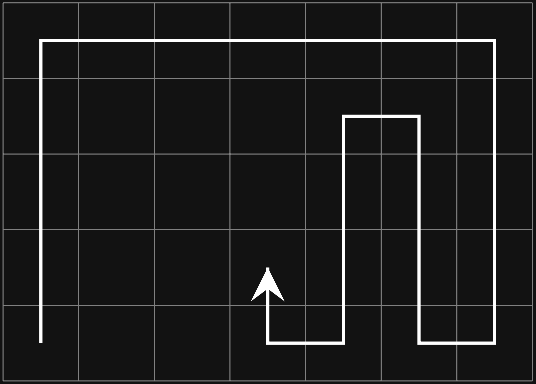
Roomba - Tilfelli 3

$n = 5$, $m = 7$



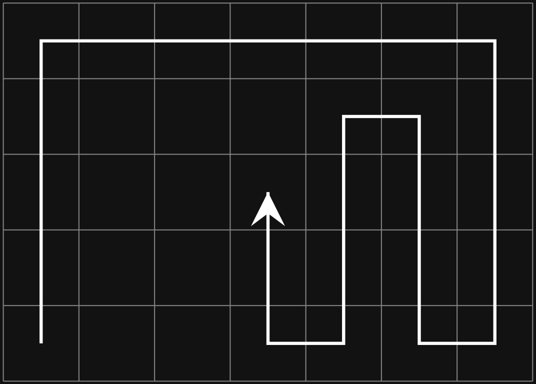
Roomba - Tilfelli 3

$n = 5, m = 7$



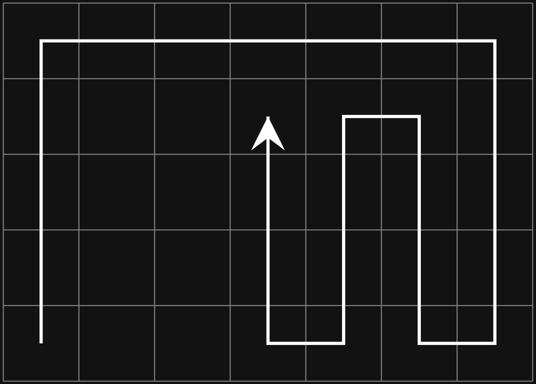
Roomba - Tilfelli 3

$n = 5$, $m = 7$



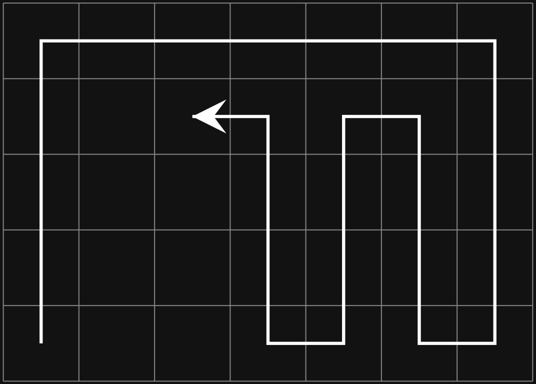
Roomba - Tilfelli 3

$n = 5$, $m = 7$



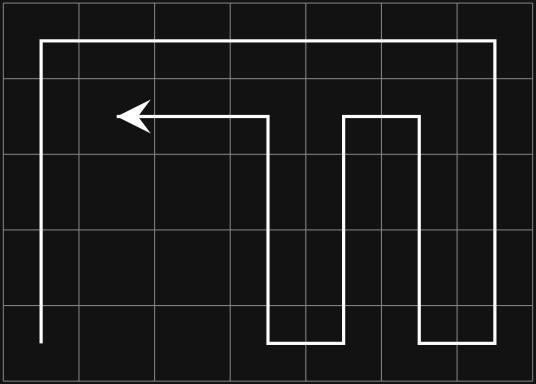
Roomba - Tilfelli 3

$n = 5, m = 7$



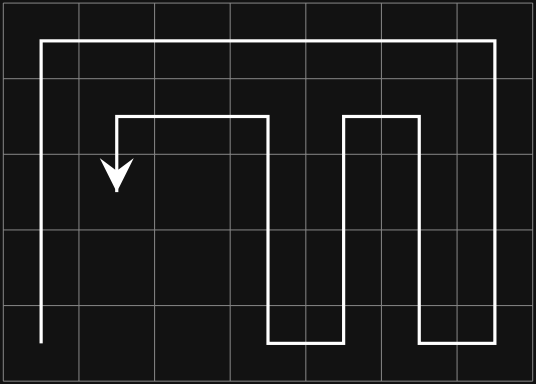
Roomba - Tilfelli 3

$n = 5$, $m = 7$



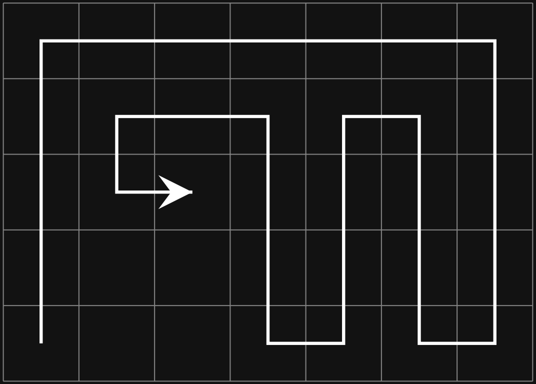
Roomba - Tilfelli 3

$n = 5$, $m = 7$



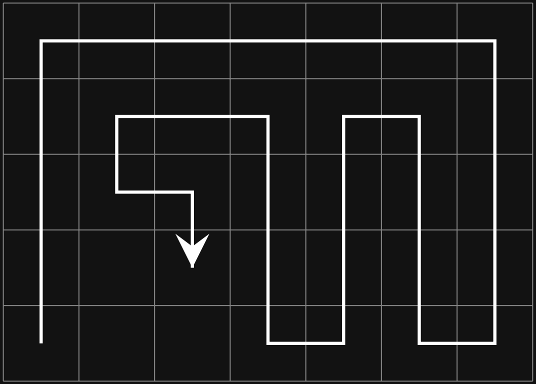
Roomba - Tilfelli 3

$n = 5$, $m = 7$



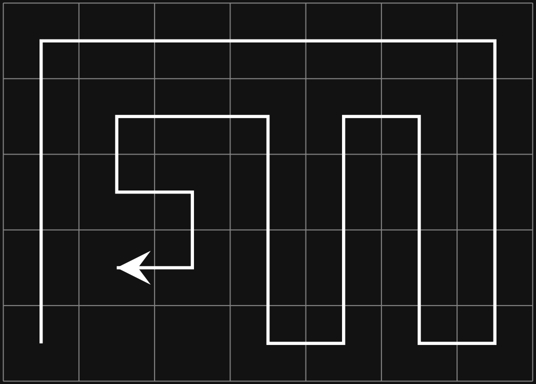
Roomba - Tilfelli 3

$n = 5$, $m = 7$



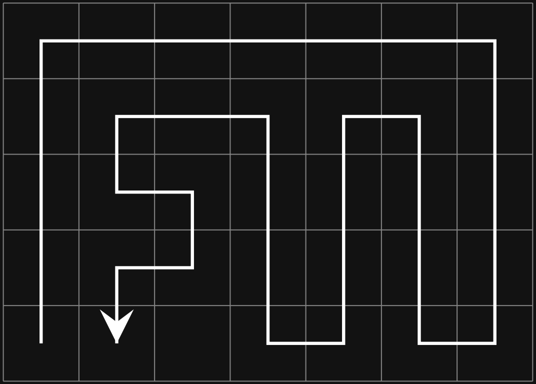
Roomba - Tilfelli 3

$n = 5, m = 7$



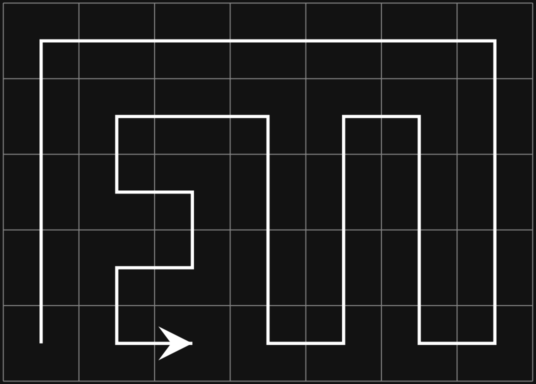
Roomba - Tilfelli 3

$n = 5$, $m = 7$



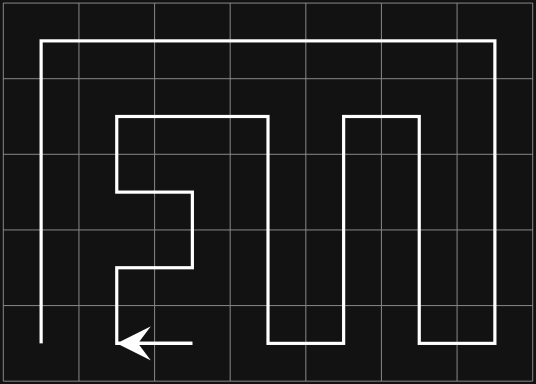
Roomba - Tilfelli 3

$n = 5, m = 7$



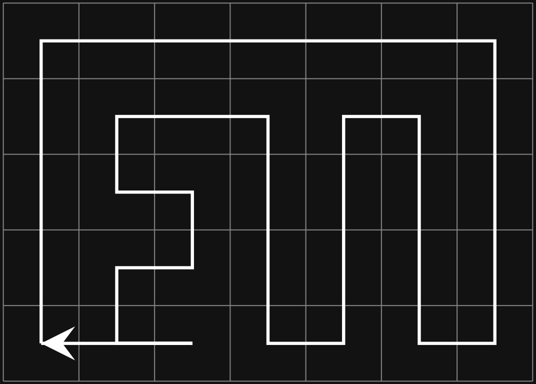
Roomba - Tilfelli 3

$n = 5$, $m = 7$



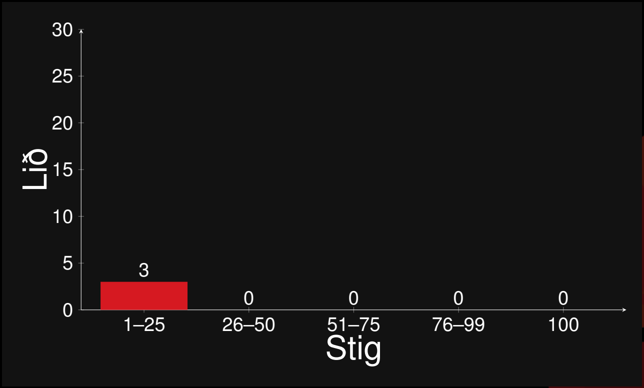
Roomba - Tilfelli 3

$n = 5$, $m = 7$



Bitaflipp

	Keppendur	Dómarar
Stysta lausn	?	34
Lengsta lausn	?	64
	Tími	Lið
Fyrsta lausn	?	?



Bitaflipp

Dæmið

Gefinn bitastrengur, „flippið“ bitunum í einhverjum hlutstreng þannig að fjölda 1-bita í útkomunni sé hámarkaður.

Lausn

- Ef 0-biti er inni í hlutstrengnum, þá mun vera einum fleiri 1-bitar í útkomunni
- Ef 1-biti er inni í hlutstrengnum, þá mun vera einum færri 1-bitar í útkomunni

1	0	0	1	0	0	1

Bitaflipp

Dæmið

Gefinn bitastrengur, „flippið“ bitunum í einhverjum hlutstreng þannig að fjölda 1-bita í útkomunni sé hámarkaður.

Lausn

- Ef 0-biti er inni í hlutstrengnum, þá mun vera einum fleiri 1-bitar í útkomunni
- Ef 1-biti er inni í hlutstrengnum, þá mun vera einum færri 1-bitar í útkomunni

1	0	0	1	0	0	1

Bitaflipp

Dæmið

Gefinn bitastrengur, „flippið“ bitunum í einhverjum hlutstreng þannig að fjölda 1-bita í útkomunni sé hámarkaður.

Lausn

- Ef 0-biti er inni í hlutstrengnum, þá mun vera einum fleiri 1-bitar í útkomunni
- Ef 1-biti er inni í hlutstrengnum, þá mun vera einum færri 1-bitar í útkomunni

1	0	0	1	0	0	1

1	1	1	0	0	0	1

Bitaflipp

Dæmið

Gefinn bitastrengur, „flippið“ bitunum í einhverjum hlutstreng þannig að fjölda 1-bita í útkomunni sé hámarkaður.

Lausn

- Ef 0-biti er inni í hlutstrengnum, þá mun vera einum fleiri 1-bitar í útkomunni
- Ef 1-biti er inni í hlutstrengnum, þá mun vera einum færri 1-bitar í útkomunni

1	0	0	1	0	0	1

1	0	0	1	1	1	1

Bitaflipp

Dæmið

Gefinn bitastrengur, „flippið“ bitunum í einhverjum hlutstreng þannig að fjölda 1-bita í útkomunni sé hámarkaður.

Lausn

- Ef 0-biti er inni í hlutstrengnum, þá mun vera einum fleiri 1-bitar í útkomunni
- Ef 1-biti er inni í hlutstrengnum, þá mun vera einum færri 1-bitar í útkomunni

1	0	0	1	0	0	1

1	1	1	0	1	1	1

Bitaflipp

Lausn

- Breytum 1-bitum í töluna -1 og 0-bitum í töluna 1

1	0	0	1	0	0	1

-1	1	1	-1	1	1	-1

- Viljum finna samliggjandi hlutrunu með hámarks summu

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Hæg lausn: Prufa allar hlutrunur og reikna summu

-1	1	1	-1	1	1	-1

```
mx = 0
for j in 1..n:
    for i in 1..i:
        sm = 0
        for k in i..j:
            sm += arr[k]
        mx = max(mx, arr[k])
print mx
```

- Getum við reiknað summuna af ákveðinni hlutrunu hratt?

Bitaflipp - Hlutruna með hámarks summu

Lausn

- Smá trikk: $\mathrm{sum}(i, j) = \mathrm{sum}(1, j) - \mathrm{sum}(1, i - 1)$

-1	1	1	-1	1	1	-1

$$=$$

-1	1	1	-1	1	1	-1

$$-$$

-1	1	1	-1	1	1	-1

- Reiknum $\mathrm{sum}(1, i)$ fyrir öll i í byrjun

Bitaflipp - Hlutruna með hámarks summu

Lausn

```
sum[0] = 0
for i in 1..n:
    sum[i] = arr[i] + sum[i-1]

mx = 0
for j in 1..n:
    for i in 1..j:
        mx = max(mx, sum[j] - sum[i-1])
print mx
```

Bitaflipp - Hlutruna með hámarks summu

Lausn

```
sum[0] = 0
for i in 1..n:
    sum[i] = arr[i] + sum[i-1]

mx = 0
for j in 1..n:
    mn = 1000000000000000000000
    for i in 1..j:
        mn = min(mn, sum[i-1])
    mx = max(mx, sum[j] - mn)
print mx
```

Bitaflipp - Hlutruna með hámarks summu

Lausn

```
sum[0] = 0
for i in 1..n:
    sum[i] = arr[i] + sum[i-1]

mx = 0
mn = 1000000000000000000000
for j in 1..n:
    mn = min(mn, sum[i-1])
    mx = max(mx, sum[j] - mn)
print mx
```

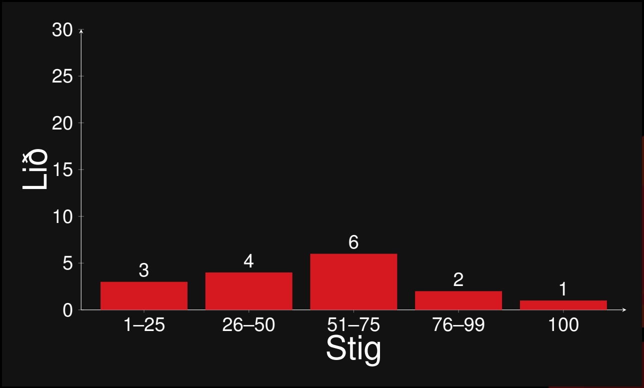
Bitaflipp

Lausn

- Passa upp á sértilfelli sem innihalda bara 1-bita

Samhverfudulritun

	Keppendur	Dómarar
Stysta lausn	25	30
Lengsta lausn	25	43
	Tími	Lið
Fyrsta lausn	1:49:48	Níels Karlsson



Samhverfudulritun

Dæmið

Fyrir hvert k á milli 1 og 100, finna stærstu samhverfu minni en 2^k.

Lausn

- Allt í lagi þó lausnin sé hæg
  - Láta forritið malla á eigin tölvu
  - Skila úttakinu beint á Kattis
- Skoða $2^k, 2^k - 1, 2^k - 2, \ldots$
- Skrifa út fyrstu samhverfu sem við finnum
- Hægt að ná rúmlega 60 stigum ef forritið fær að malla nógu lengi
- Of hægt fyrir stór k

Samhverfudulritun

Lausn

- Tökum $k = 92$ sem dæmi

$$\begin{array}{c}4951760157141521099596496896 \\ 4951760157141441417510671594\end{array}$$

Samhverfudulritun

Lausn

- Tökum $k = 92$ sem dæmi

49517601571415	21099596496896
49517601571414	41417510671594

Samhverfudulritun

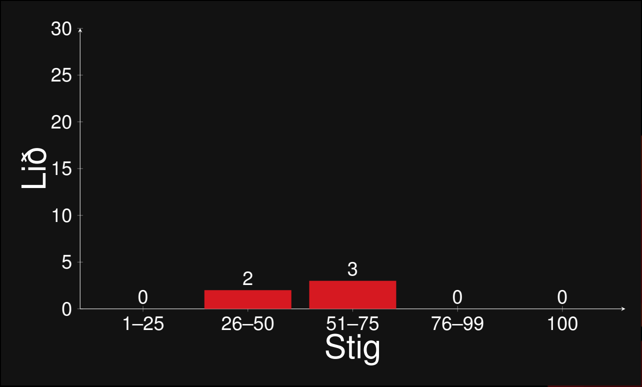
Lausn

- Tökum $k = 92$ sem dæmi

495176015714155	21099596496896
495176015714144	41417510671594

Dans

	Keppendur	Dómarar
Stysta lausn	?	19
Lengsta lausn	?	56
	Tími	**Lið**
Fyrsta lausn	?	?



Dans

Dæmið

Gefin gagntæk vörpun f, reikna $f^{(k)}$.

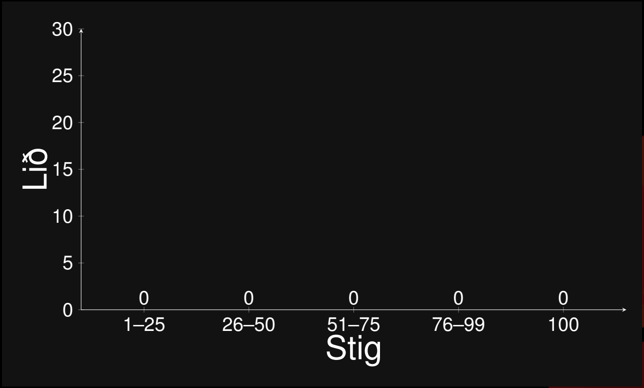
Lausn

- Hægt að horfa á danshreyfingarnar sem áttað net
- Tökum 3 5 1 2 4 sem dæmi

- Netið mun eingöngu samanstanda af svona rásum
- Manneskja i byrjar í hnút i og fer svo yfir k leggi

Bolir

	Keppendur	Dómarar
Stysta lausn	?	33
Lengsta lausn	?	52
	Tími	Lið
Fyrsta lausn	?	?



Bolir

Dæmið

Úthluta bolum til keppenda þannig að hver keppandi fái bol sem er hvorki of stór né of lítill.

Lausn

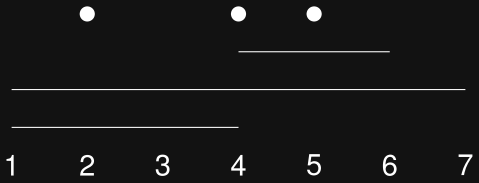
- Höfum tvær gerðir af hlutum
- Pör af hlutum af mismunandi gerðum geta „passað saman“
- Viljum para saman hluti af mismunandi gerðum, þannig að allir paraðir hlutir „passi saman“

- Þetta kallast spyrðing í tvíhlutaneti (e. bipartite matching)
  - Til skilvirk reiknirit, t.d. með netaflæði
  - Fjöldi keppenda og bola er mjög stór, svo þetta er of hægt

Bolir

Lausn

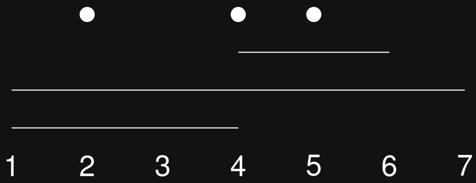
- Aftur gott að setja þetta upp grafískt:
  - Bolur af stærð x táknaður með punkti í x
  - Keppandi sem vill stærð frá y til z táknaður með línustriki frá y til z



Bolir

Lausn

- Göngum á bolinu frá vinstri til hægri
- Af þeim keppendum sem geta fengið þennan bol, tökum þann sem hefur vinstrasta hægri endapunkt

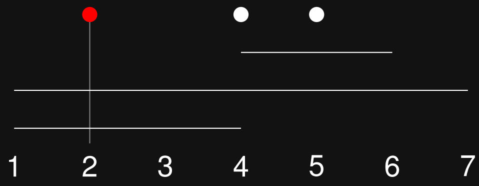


- Nota forgangsbiðröð (e. priority queue) til að útfæra í $O(n \log n)$ tíma

Bolir

Lausn

- Göngum á bolinu frá vinstri til hægri
- Af þeim keppendum sem geta fengið þennan bol, tökum þann sem hefur vinstrasta hægri endapunkt

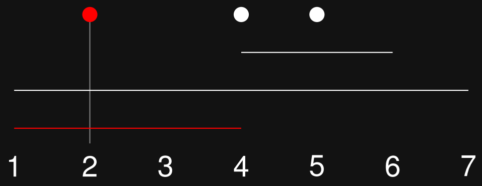


- Nota forgangsbiðröð (e. priority queue) til að útfæra í $O(n \log n)$ tíma

Bolir

Lausn

- Göngum á bolinu frá vinstri til hægri
- Af þeim keppendum sem geta fengið þennan bol, tökum þann sem hefur vinstrasta hægri endapunkt

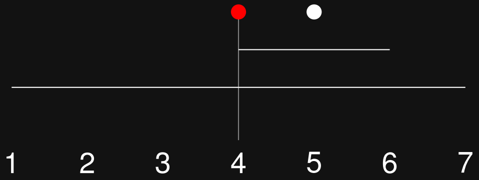


- Nota forgangsbiðröð (e. priority queue) til að útfæra í $O(n \log n)$ tíma

Bolir

Lausn

- Göngum á bolinu frá vinstri til hægri
- Af þeim keppendum sem geta fengið þennan bol, tökum þann sem hefur vinstrasta hægri endapunkt

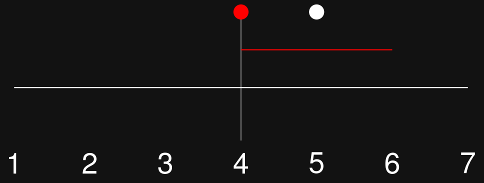


- Nota forgangsbiðröð (e. priority queue) til að útfæra í $O(n \log n)$ tíma

Bolir

Lausn

- Göngum á bolinu frá vinstri til hægri
- Af þeim keppendum sem geta fengið þennan bol, tökum þann sem hefur vinstrasta hægri endapunkt

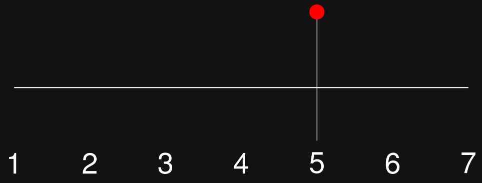


- Nota forgangsbiðröð (e. priority queue) til að útfæra í $O(n \log n)$ tíma

Bolir

Lausn

- Göngum á bolinu frá vinstri til hægri
- Af þeim keppendum sem geta fengið þennan bol, tökum þann sem hefur vinstrasta hægri endapunkt



- Nota forgangsbiðröð (e. priority queue) til að útfæra í $O(n \log n)$ tíma

Bolir

Lausn

- Göngum á bolinu frá vinstri til hægri
- Af þeim keppendum sem geta fengið þennan bol, tökum þann sem hefur vinstrasta hægri endapunkt

- Nota forgangsbiðröð (e. priority queue) til að útfæra í $O(n \log n)$ tíma

Bolir

Lausn

- Göngum á bolinu frá vinstri til hægri
- Af þeim keppendum sem geta fengið þennan bol, tökum þann sem hefur vinstrasta hægri endapunkt

1 2 3 4 5 6 7

- Nota forgangsbiðröð (e. priority queue) til að útfæra í $O(n \log n)$ tíma

Skemmtileg tölfræði

- Minnsti fjöldi lína sem þarf til að leysa öll Commodore 64: 406
- Fjöldi committa í Git repositoryinu okkar: 233
- Heildarfjöldi lína í öllum skrám sem við koma verkefnunum: 4828758
- Fjöldi stafsetningavilla í fyrirspurnum sem Hjalti sér eftir: 1
- Fjöldi dæma sem voru einfaldari en við héldum: 1