



Tölvunarfræði 1

Fyrirlestur 8: Lykkjur II

Hjálmtyr Hafsteinsson
Haust 2015



HÁSKÓLI ÍSLANDS

ÍDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Í síðasta fyrirlestri

- Helstu gerðir lykkja í Java
- **while**-lykkjan
 - Ítra á meðan skilyrði satt
- **for**-lykkjan
 - Ítra í tiltekinn fjölda skipta

Kafli 1.3



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Í þessum fyrirlestri

- Dæmi um notkun á lykkjum
 - Gerðir lykkja
 - Að hoppa út úr lykkjum
- Hreiðraðar lykkjur
 - Lykkjur með **if**-setningum

Kafli 1.3



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD

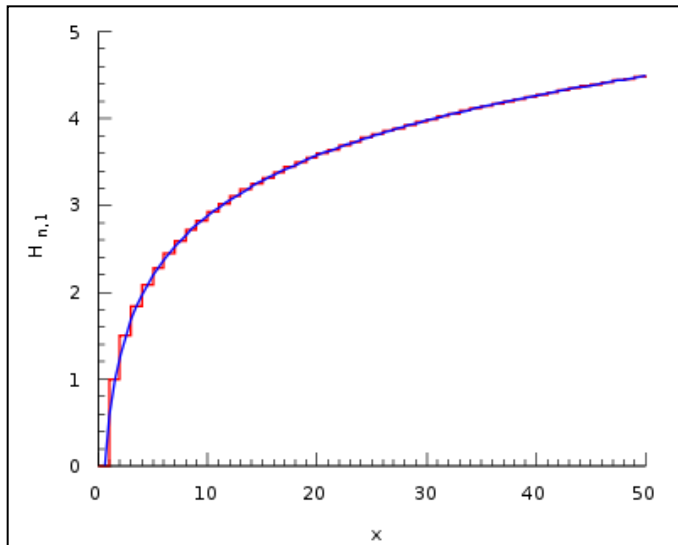


Notkun lykkja

- Reikna gildi stærðfræðilegra raða

Dæmi: Reikna gildi n -tu pýðtölu (*Harmonic number*)

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} = \sum_{k=1}^n \frac{1}{k}.$$



([Java Visualizer](#))

```
public class Harmonic {  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
  
        double sum = 0.0;  
        for (int i = 1; i <= N; i++) {  
            sum += 1.0 / i;  
        }  
  
        System.out.println(sum);  
    }  
}
```



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Nálgun á kvaðratrót

- Dýrt að reikna kvaðratrót
- Notum nálgunaraðferð Newton-Raphson:

Finna kvaðratrót tölunnar c :

Upphafsstilla t_0 sem c

Á meðan ennþá er munur á t_i og c/t_i

Setja t_{i+1} sem meðaltal t_i og c/t_i

Almenn aðferð til að finna núllstöðvar falla

Nokkuð hröð samleitni.
Mikið notuð í reynd

t_0	=	2.0
t_1	= $\frac{1}{2}(t_0 + \frac{2}{t_0})$	= 1.5
t_2	= $\frac{1}{2}(t_1 + \frac{2}{t_1})$	= 1.4166666666666665
t_3	= $\frac{1}{2}(t_2 + \frac{2}{t_2})$	= 1.4142156862745097
t_4	= $\frac{1}{2}(t_3 + \frac{2}{t_3})$	= 1.4142135623746899
t_5	= $\frac{1}{2}(t_4 + \frac{2}{t_4})$	= 1.414213562373095



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Nálgun á kvaðratrót, kóðinn

([Java Visualizer](#))

```
public class Sqrt {  
    public static void main(String[] args) {  
        double c = Double.parseDouble(args[0]);  
        double epsilon = 1e-15;  
        double t = c;  
        int count = 0;  
  
        while (Math.abs(t - c/t) > epsilon*t) {  
            t = (c/t + t) / 2.0;  
            count++;  
        }  
  
        System.out.println("Iterations: " + count);  
        System.out.println(t);  
    }  
}
```

Tala til að finna kvaðratrót af

Hversu mikla nákvæmni viljum við?

Á meðan munurinn á t og c/t er stærri en ϵ ...

```
% java Sqrt 2  
Iterations: 5  
1.414213562373095
```

```
% java Sqrt 20000  
Iterations: 12  
141.4213562373095
```

Fjöldi ítrana er lágur, en háður stærð tölunnar c



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Breyta milli talnagrunna

- Breyta úr tugatölu í tvíunartölu (*binary*)
 - Finnum fyrst næsta heila veldið á 2 fyrir neðan töluna
 - Rekjum okkur svo niður veldin á 2

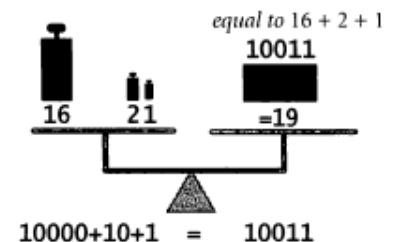
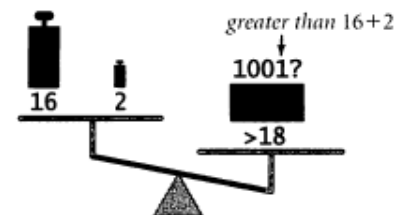
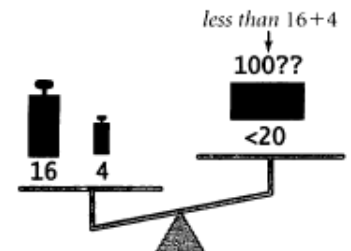
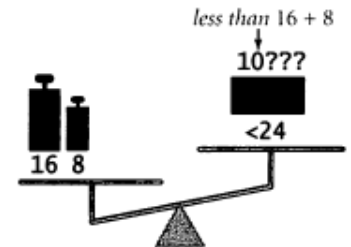
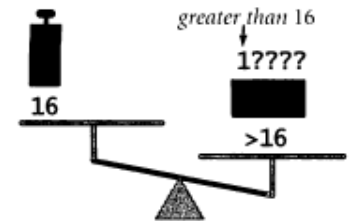
Ef $N = 19$, þá er það 16 (2^4) $\Rightarrow 1????$

Prófum 8 (2^3), en $16+8 = 24$ sem er stærra en 19 $\Rightarrow 10???$

Prófum 4 (2^2), en $16+4 = 20$ sem er stærra en 19 $\Rightarrow 100??$

Prófum 2 (2^1), og $16+2 = 18$ sem er ekki stærra en 19 $\Rightarrow 1001?$

Prófum 1 (2^0), og $16+2+1 = 19$ sem er ekki stærra en 19 $\Rightarrow 10011$



Scale analog to binary conversion



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Breyta í tvíundartölur (Java kóði)

(Java Visualizer)

Finna fyrst stærsta heila veldi af 2 sem er $\leq n$

Ef heila veldið af 2 of stórt, þá kemur 0, annars 1

Rekjum okkur niður heilu veldin af 2 með því að deila með 2 uppí v

```
int v = 1;
while (v <= n/2) {
    v = v * 2;
}

while (v > 0) {
    if (n < v) {
        System.out.print(0);
    }
    else {
        System.out.print(1);
        n = n - v;
    }
    v = v / 2;
}
System.out.println();
```



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



break skipunin

- Stundum viljum við komast út úr lykkju án þess að skilyrði lykkjunnar sé orðið ósatt
 - Stundum eitthvert sértilfalli sem kemur sjaldan upp
 - Viljum ekki flækja lykkjuskilyrðið með því að athuga það þar
- Notum þá **break** skipunina
 - Höfum séð hana notaða í **switch**-setningunni
 - Hér höfum við yfirleitt **if**-setningu á undan henni



Dæmi um notkun break

- Athuga hvort gefin heiltala N sé frumtala (*prime*)

```
public class TestPrime {  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
        int i;  
  
        for (i = 2; i <= N; i++) {  
            if (N % i == 0) break;  
        }  
        if (i > N/i) {  
            System.out.println(N + " is prime");  
        }  
    }  
}
```

Prófum allar tölur $\leq N$.
Ef ein þeirra gengur
upp í N , þá hætta

```
% java TestPrime 5  
5 is prime  
  
% java TestPrime 6  
  
% java TestPrime 123
```

Skrifar ekkert ef
talan er samsett



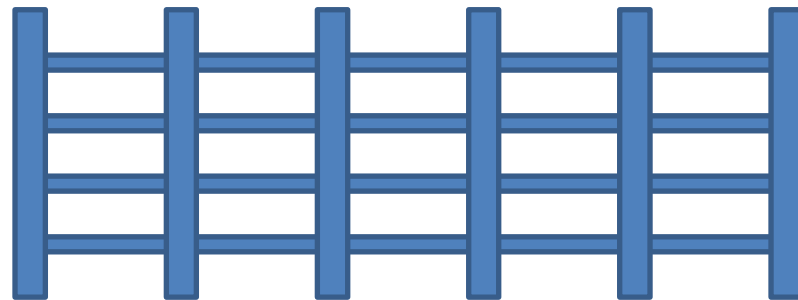
HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Girðingastauralykkja (*fence post loop*)

- Í mörgum tilvikum þurfum við aukaskipun á undan (eða eftir) lykkju til að leysa gefið verkefni
 - Þegar við smíðum girðingu með n bilum (gaddavír), þá þurfum við $n+1$ staur
 - Ef við notum bara eina lykkju sem ítrar n sinnum þá fáum við aðeins n staura og n bil



Þurfum sér skipun til að
búa til þennan staur



Girðingastauralykkja - dæmi

- Skrifa út heiltölurnar 1 til N með kommu á milli
 - Til dæmis: 1, 2, 3, 4, 5, 6

([Java Visualizer](#))

```
System.out.print(1);  
for (int i = 2; i <= N; i++) {  
    System.out.print(", " + i);  
}  
System.out.println();
```

Fyrsti staurinn utan lykkju

Til að fara í nýja línu

eða

```
for (int i = 1; i <= N-1; i++) {  
    System.out.print(i + ", ");  
}  
System.out.println(N);
```

Síðasti staurinn utan lykkju og farið í nýja línu



Fyrirlestraræfing

1. Sýnið **for**-lykkju til að reikna fyrstu N liðina í röðinni $\frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \dots + \frac{1}{4^N}$
2. Hvernig er tugatalan 37 skrifuð sem tvíundartala?
3. Sýnið **for**-lykkju sem skrifar út "girðingu" með N staurum eins og sést hér fyrir neðan

| = | = | = | = | = | = |





Hreiðraðar (*nested*) lykkjur

- Getum haft lykkjur inni í lykkjum
 - Fyrir hverja ítrun ytri lykkju framkvæmist öll innri lykkjan

Dæmi: Prenta út kassa af stjörnum

([Java Visualizer](#))

```
public class StjornuKassi {  
    public static void main(String[] args) {  
        int n = Integer.parseInt(args[0]);  
        int m = Integer.parseInt(args[1]);  
  
        for (int i=0; i<n; i++) {  
            for (int j=0; j<m; j++) {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
    }  
}
```

```
% java StjornuKassi 5 10  
*****  
*****  
*****  
*****  
*****
```



Dæmi: Deilingarmynstur

([Java Visualizer](#))

```
public class DivisorPattern {  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
  
        for (int i = 1; i <= N; i++) {  
            for (int j = 1; j <= N; j++) {  
                if (i % j == 0 || j % i == 0) {  
                    System.out.print("* ");  
                } else {  
                    System.out.print("  ");  
                }  
            }  
            System.out.println(i);  
        }  
    }  
}
```

Býr til töflu yfir það
hvaða tölur ganga
upp í hverja aðra

```
% java DivisorPattern 8  
* * * * * * * 1  
* *   *   *   * 2  
*   *       *   3  
* *   *       * 4  
*       *       5  
* * *       *   6  
*       *       7  
* *   *       * 8
```

Stjarna í (i, j) ef i
gengur upp í j
eða öfugt



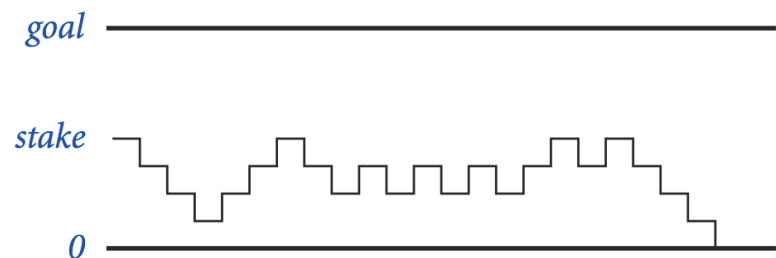
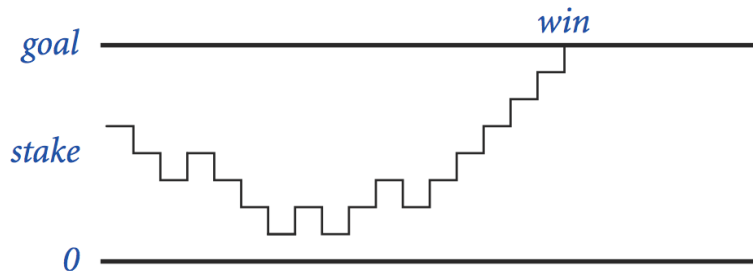
HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Hermun (*simulation*)

- Glötun fjárhættuspilarans (*Gambler's ruin*)
 - Spilari byrjar með $\$stake$ pening
 - Veðjar $\$1$ í hvert sinn (50% líkur á vinningi)
 - Vinnur ef hann nær $\$goal$ peningum
 - Tapar ef hann klárar peninginn
- Ýmsar spurningar:
 - Hverjar eru líkurnar á að hann vinni?
 - Hversu mörg veðmál þarf þar til leikur búinn?





Einn leikur

```
int bets = 0;
int cash = stake;
while (cash > 0 && cash < goal) {
    bets++;
    if (Math.random() < 0.5)
        cash++;      // win $1
    else
        cash--;      // lose $1
}
```

```
if (cash == goal)
    System.out.println("You won!");
```

Á meðan ekki tap (`cash > 0`)
og ekki sigur (`cash < goal`)

Telja fjölda veðmála

Varpa hlutkesti

Unnum veðmál

Töpuðum veðmáli

Athuga útkomu, þ.e. af hverju
hættum við í `while`-lykkju?

Fáum mjög ólíkar niðurstöður
úr keyrslum forritsins

Hugmynd:
Keyrum oft og
reiknum meðaltöl



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Hermun á glötun fjárhættuspilarans

([Java Visualization](#))

```
public class Gambler {  
    public static void main(String[] args) {  
        int stake = Integer.parseInt(args[0]);  
        int goal  = Integer.parseInt(args[1]);  
        int T     = Integer.parseInt(args[2]);  
        int bets  = 0;  
        int wins  = 0;  
  
        for (int t = 0; t < T; t++) {  
            int cash = stake;  
            while (cash > 0 && cash < goal) {  
                bets++;  
                if (Math.random() < 0.5) cash++;  
                else cash--;  
            }  
            if (cash == goal) wins++;  
        }  
        System.out.println(wins + " wins of " + T);  
        System.out.println("Percent of games won = " + 100.0*wins/T);  
        System.out.println("Avg # bets          = " + 1.0*bets/T);  
    }  
}
```

Fáum nú líka inn
fjölda tilrauna T

Einn leikur (þ.e. tilraun)

Fylgjumst nú með
fjölda sigra

Skrifum út ýmsar
tölfræðiniðurstöður



Niðurstöður hermunar

stake = 20
goal = 100
T = 10000

```
% java Gambler 20 100 10000
1952 wins of 10000
Avg # bets          = 1574.6278

% java Gambler 20 100 10000
2017 wins of 10000
Avg # bets          = 1618.114

% java Gambler 20 100 10000
2042 wins of 10000
Avg # bets          = 1651.6438
```

Stærðfræðin
segir:

- Líkur á sigri í hverju leik eru *stake/goal*
 - Hér er það $20/100 = \underline{20\%}$
- Væntur fjöldi veðmála er *stake * (goal - stake)*
 - Hér er það $20 * (100 - 20) = 20 * 80 = \underline{1600}$

Passar nokkuð
vel við okkar
niðurstöður

Getur verið mjög erfitt að sanna
eitthvað um flóknari útgáfur.
Hermun er þá lausnin!



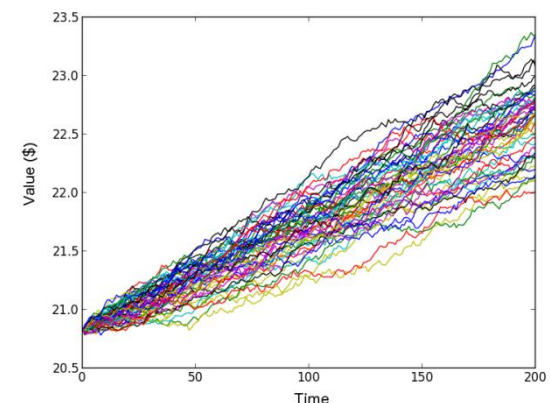
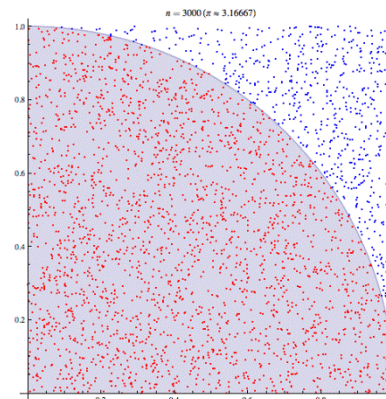
HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Almennt um hermun

- Notað þegar erfitt (eða ómögulegt) er að nota stærðfræðilegar aðferðir
 - Kallast Monte Carlo hermun þegar við notum slembitölur við hermunina (sem er oftast!)
 - Valin ein af 10 mikilvægustu reikniritum 20. aldarinnar
 - Mörg notkunar svið:
 - Hegðun verðbréfamarkaða
 - Hermun frumeinda samkvæmt lögmálum skammtafræðinnar
 - Nálgun heildunarverkefna
 - Veðurfræði
 - Þarf að hafa í huga:
 - Gæði líkans
 - Gæði slembitalna!



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐI DEILD



do-while skipunin

- Þriðja gerð lykkja í Java
 - Framkvæmist alltaf a.m.k. einu sinni
 - Stundum kölluð **repeat-until**

```
do {  
    setning;  
} while (röksegð)
```

- Hentar í vissum aðstæðum:

Gera eitthvað, ef það gekk ekki, gera aftur!

- Dæmi:
 - Fá inntak frá notanda, þangað til það er löglegt
 - Fá slembigildi með tiltekna eiginleika





do-while dæmi

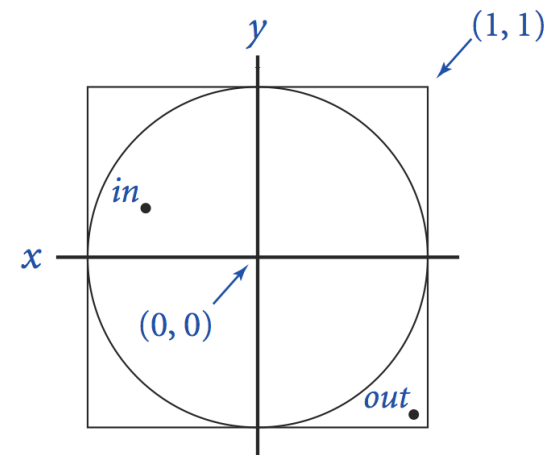
- Finna slembitölur x og y , þannig að hnitin (x, y) séu jafndreifð á einingarskífunni (*unit disk*)
 - Finna x og y , endurtaka þar til (x, y) er innan einingarskífu

([Java Visualizer](#))

```
double x, y;  
do {  
    x = 2.0*Math.random() - 1.0;  
    y = 2.0*Math.random() - 1.0;  
} while (Math.sqrt(x*x + y*y) > 1.0);
```

Láta x og y vera á bilinu $[-1, 1]$

Vitum ekki alveg hversu oft við þurfum að ítra





Fyrirlestraræfing

4. Í síðustu `println`-skipun forritsins `Gambler` er segðin `1.0*bets/T`. Hvers vegna er margfaldað með `1.0`?
5. Í `Gambler` leiknum, ef þið byrjið með \$10 og viljið hætta með \$100 hvað megið þið búast við að þurfa að veðja oft?
6. Skrifið `do-while` setningu sem notar `Math.random()` til að finna slembitölu á bilinu (0, 0.5)





Samantekt

- Í þessum tíma:
 - Dæmi um notkun lykkja
 - Hreiðraðar lykkjur
- Í næsta tíma:
 - Fylki (*arrays*) í Java
 - Dæmi um notkun

Kafli 1.3

Kafli 1.4



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD