



Tölvunarfræði 1

Fyrirlestur 9: Fylki I

Hjálmtyr Hafsteinsson
Haust 2015



HÁSKÓLI ÍSLANDS

ÍDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Í síðasta 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



Í þessum fyrirlestri

- Fylki (*arrays*) í Java
 - Skilgreining
 - Minnisúthlutun
- Dæmi um notkun á fylkjum:
 - Að stokka spil (*shuffle*)
 - Safnaraverkefnið

Kafli 1.4



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Fylki (*arrays*) í Java

- Notuð til að halda utanum mikið magn gagna

Dæmi:

- 52 spil í spilastokk
- 360 nemendur í þessu námskeiði
- 1 milljón stafir í bók
- 10 milljón hljóðmerki í MP3 skrá
- 3 milljarðar basapara í DNA
- [1.2 milljón milljónir](#) fyrirspurna hjá Google á ári

vísir	gildi
0	hh
1	snorri
2	pmelsted
3	johann
4	jonasson
5	helmut
6	ebba
7	steinng

Skilgreining:

Fylki er runa gagna af sama tagi



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Hvers vegna þurfum við fylki?

- Höfum 10 nemendur í námskeiði, viljum geyma einkunnir þeirra

Skilgreina eina breytu fyrir hvern nemanda

Þurfum að upphafsstillastær allar sjálfstætt

```
// þreytandi, auðvelt að gera villur
double a0, a1, a2, a3, a4, a5, a6, a7, a8, a9;
a0 = 0.0;
a1 = 0.0;
a2 = 0.0;
a3 = 0.0;
a4 = 0.0;
a5 = 0.0;
a6 = 0.0;
a7 = 0.0;
a8 = 0.0;
a9 = 0.0;

...
a4 = 3.0;

...
a8 = 8.0;

...
double x = a4 + a8;
```



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Hvers vegna þurfum við fylki?

- Höfum 10 nemendur í námskeiði, viljum geyma einkunnir þeirra

Skilgreina eitt fylki fyrir
alla nemendurna

Fylkið er sjálfkrafa
upphafsstillt (sem 0.0)

```
// betri leið  
double[] a = new double[10];  
...  
a[4] = 3.0;  
...  
a[8] = 8.0;  
...  
double x = a[4] + a[8];
```

Sjáum á eftir hvernig
þessi skipun virkar



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Hvers vegna þurfum við fylki?

- Höfum 1,000,000 gildi sem við viljum geyma

Ómögulegt að búa til 1 milljón sjálfstæðar breytur

Notum `int` til að vísa í fylki. Hámarksstærð fylkja takmarkast af því

```
// Einfalt að búa til stór fylki
double[] a = new double[1000000];

...
a[123456] = 3.0;

...
a[987654] = 8.0;

...
double x = a[123456] + a[987654];
```

Fylki gefa okkur leið til að vinna með mikið gagnamagn á skýran og einfaldan hátt



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Skilgreining fylkja

- Fylki eru innbyggð í Java
 - Búa til fylki: skilgreina, skapa og upphafsstill
 - Nota stak númer i í fylki a : notum $a[i]$
 - Fyrsta stakið er númer 0

```
int N = 10;           // stærð fylkis
double[] a;          // skilgreina fylkið
a = new double[N];    // skapa fylkið
for (int i = 0; i < N; i++) // upphafsstill fylkið
    a[i] = 0.0;       // allt sem 0.0
```

- Einfaldari lausn:
 - Skilgreina og skapa í sömu setningu
 - Java núllstillir fyrir okkur (þ.e. setur sem 0.0)

```
int N = 10;           // stærð fylkis
double[] a = new double[N]; // skilgr., skapa, upph.st.
```





Að búa til fylki

- Stytttri aðferðin við að búa til fylki:

```
int N = 10;  
int[] a = new int[N];
```

- Núllstilling fer eftir tagi stakanna
 - Ef `int` þá sett 0
 - Ef `double` þá sett 0.0
 - Ef `boolean` þá sett `false`
 - Ef `string` þá sett `null` (þ.e. enginn strengur)
- Núllstilling tekur tíma í hlutfalli við stærð fylkis
 - Ekki ókeypis!





Notkunardæmi

- Reikna innfeldi (*dot product*) fylkjanna $\mathbf{x}[]$ og $\mathbf{y}[]$
 - Leggja saman margfeldi samsvarandi staka

```
double[] x = { 0.3, 0.6, 0.1 };  
double[] y = { 0.5, 0.1, 0.4 };  
int N = x.length;  
double sum = 0.0;  
  
for (int i = 0; i < N; i++) {  
    sum = sum + x[i]*y[i];  
}
```

Önnur leið til að búa til fylki: Telja upp gildin

Til að fá lengd fylkisins

i	x[i]	y[i]	x[i]*y[i]	sum
				0
0	.30	.50	.15	.15
1	.60	.10	.06	.21
2	.10	.40	.04	.25
				.25



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Nokkur einföld dæmi

([Java Visualizer](#))

Búa til fylkið `a` og setja slembigildi í það

```
double[] a = new double[N];  
for (int i = 0; i < N; i++)  
    a[i] = Math.random();
```

Prenta út gildi fylkisins eitt stak í línu

```
for (int i = 0; i < N; i++)  
    System.out.println(a[i]);
```

Finna stærsta stak fylkisins

```
double max = Double.NEGATIVE_INFINITY;  
for (int i = 0; i < N; i++)  
    if (a[i] > max)  
        max = a[i];
```



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Fleiri einföld dæmi

Finna meðaltal gildanna í fylkinu *a*

```
double sum = 0.0;
for (int i = 0; i < N; i++)
    sum += a[i];
double average = sum / N;
```

Afrita fylkið *a* yfir í fylkið *b*

```
double[] b = new double[N];
for (int i = 0; i < N; i++)
    b[i] = a[i];
```

Snúa stökum fylkisins *b* við. Fremsta verður aftast, o.s.frv.

```
for (int i = 0; i < N; i++) {
    double temp = b[i];
    b[i] = b[N-i-1];
    b[N-i-1] = temp;
}
```



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Fyrirlestraræfing

1. Hvað er rangt við eftirfarandi forrit?

```
int[] a;  
for (int i = 0; i < 10; i++)  
    a[i] = i * i;
```

2. Hvert er innihald fylkisins **b** eftir keyrslu þessa forritsbúts?

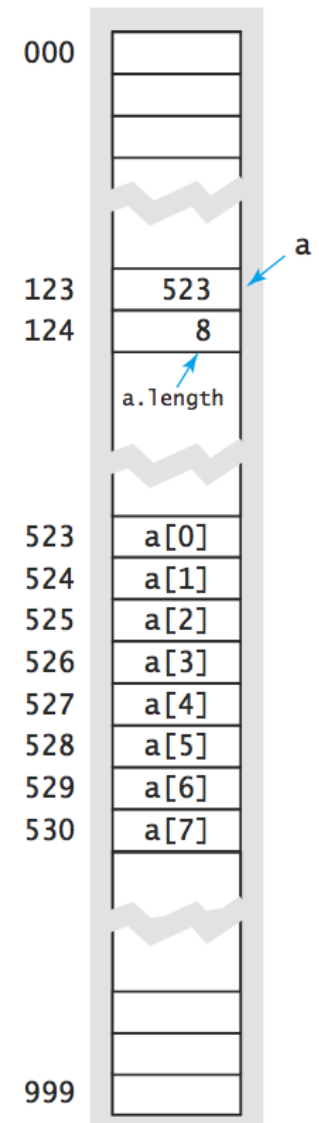
```
boolean[] b = new boolean[N];  
for (int i = 0; i < N; i++)  
    b[i] = (i%2 == 0);
```

3. Skrifið forritsbút sem býr til 10 staka heiltölufylkið **a** og upphafsstillir það með gildunum 0, 1, ..., 9



Minnisskipulag fylkja

- Fylki eru geymd í samfelldum minnishólfum
 - Fyrir hvert fylki er geymd staðsetning og stærð
 - Við skilgreiningu er búið til hólf fyrir *a* og stærð fylkisins (hér hólf 123 og 124)
 - Við sköpun (*creation*) er úthlutað minnissvæði af réttri stærð (hér 523-530)
 - Við upphafsstillingu eru sett gildi í hólfín (vitum ekki hvað var í þeim fyrir)
- Númer hólfanna eru í raun **hliðrun** (*offset*) frá fyrsta sætinu



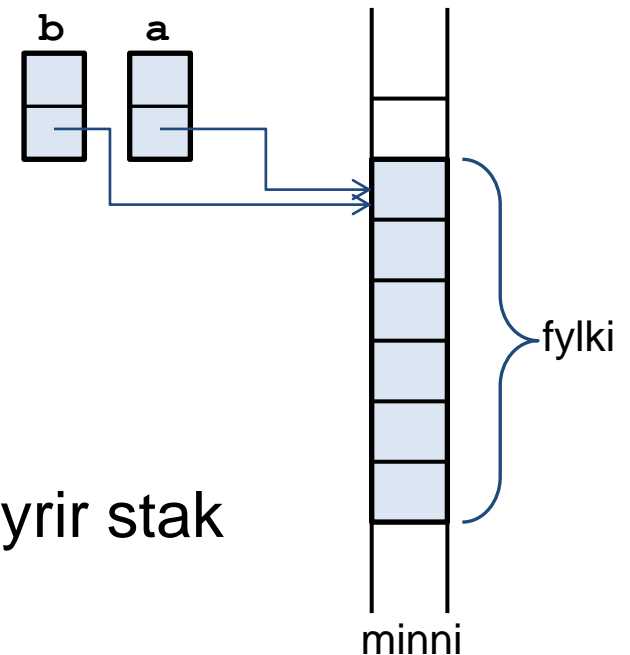


Afleiðing minnisskipulags

- Bein gildisveiting virkar ekki rétt

```
int[] a = new int[10];  
int[] b;  
  
b = a;
```

- Fylkið **b** verður sama fylkið og **a**
- Þegar **a** er breytt, þá breytist **b**, og öfugt
- Þurfum að úthluta **b** minni og afrita stak fyrir stak



- Ekki hægt að prenta fylki beint út

```
System.out.println(a);
```

```
% java Fylki  
[I@582edf43
```

Úttakið er númerið á minnishólfi **a[0]**





Spilastokkur sem fylki

Gildi sett í fylkin við
þýðingu forritsins

```
String[] rank = {"2", "3", "4", "5", "6", "7", "8", "9",  
                "10", "Jack", "Queen", "King", "Ace"};  
  
String[] suit = {"Clubs", "Diamonds", "Hearts", "Spades"};  
  
int i = (int) (Math.random() * 13);    // between 0 and 12  
int j = (int) (Math.random() * 4);    // between 0 and 3  
  
System.out.println(rank[i] + " of " + suit[j]);
```

Velja slembigildi og
slembilit



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Gildi sett í fylki í keyrslu

- Setjum oftast gildi í fylki þegar forritið er keyrt
 - Nauðsynlegt fyrir stór fylki

```
String[] deck = new String[52];  
for (int i = 0; i < 13; i++)  
    for (int j = 0; j < 4; j++)  
        deck[4*i + j] = rank[i] + " of " + suit[j];  
  
for (int i = 0; i < 52; i++)  
    System.out.println(deck[i]);
```

Spilastokkur hefur 52 stök, fyllum hann í lykkju (þ.e. í keyrslu)

```
int SUITS = suit.length;  
int RANKS = rank.length;  
int N = SUITS * RANKS;
```

Betra að skilgreina fasta með skiljanlegum nöfnum og nota þá í stað talna



Stokkun (*shuffle*)

- Viljum stokka spilin
 - Setja þau í slembiröð, þar sem sérhver möguleg uppröðun er jafnlíkleg
- Ekki eins auðvelt og það virðist við fyrstu sýn
 - Einföld aðferð:
 - Fyrir öll 52 spilin:
 - Víxla því við slembispil úr öllum stokknum
 - Vandamál:
 - Sumar uppraðanir eru líklegri en aðrar





Fisher-Yates stökkun [\(Wikipedia\)](#)

- Í ítrun i : velja af handahófi stak úr $deck[i]$ til $deck[N-1]$
- Víxla því við $deck[i]$

```
int N = deck.length;
for (int i = 0; i < N; i++) {
    int r = i + (int)(Math.random() * (N-i));
    String t = deck[r];
    deck[r] = deck[i];
    deck[i] = t;
}
```

Búa til slembi-
heiltölu frá i til $N-1$

Víxla á $deck[i]$ og
 $deck[r]$



```
public class Deck {  
    public static void main(String[] args) {  
        String[] suit = { "Clubs", "Diamonds", "Hearts", "Spades" };  
        String[] rank = { "2", "3", "4", "5", "6", "7", "8", "9",  
                           "10", "Jack", "Queen", "King", "Ace" };  
  
        int SUITS = suit.length;  
        int RANKS = rank.length;  
        int N = SUITS * RANKS;  
  
        String[] deck = new String[N];  
        for (int i = 0; i < RANKS; i++)  
            for (int j = 0; j < SUITS; j++)  
                deck[SUITS*i + j] = rank[i] + " of " + suit[j];  
  
        for (int i = 0; i < N; i++) {  
            int r = i + (int) (Math.random() * (N-i));  
            String t = deck[r];  
            deck[r] = deck[i];  
            deck[i] = t;  
        }  
  
        for (int i = 0; i < N; i++)  
            System.out.println(deck[i]);  
    }  
}
```

Skilgreina spilin

Skilgreina fasta

Búa til spilastokk

Stokka spilin

Prenta út öll spilin



Safnaraverkefnið (*coupon collector*)

- Það eru N ólíkar gerðir af spilum, hvað þarftu að safna mörgum þar til þú hefur a.m.k. eitt spil af hverri gerð?
 - Fótboltamyndir
 - Hönnun flýtiminna (*cache memory*)
 - Sannreyna gæði slembitölugjafa

- Lausnaraðferðir:



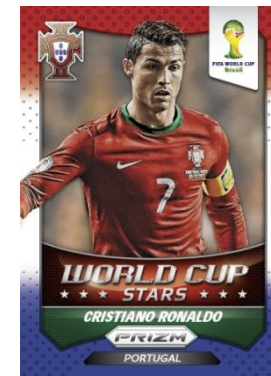
- Hermun

- Skrifa Java forrit og safna tölfræði



- Stærðfræðileg greining

- Finna lokaða formúlu fyrir væntum fjölda sem þarf að safna



HÁSKÓLI ÍSLANDS

IDNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD



Hermun á safnaraverkefni

- Velja slembiheiltölu á milli 0 og $N-1$
- Gera þetta þangað til allar tölurnar frá 0 til $N-1$ hafa komið
- Hvernig vitum við hvenær allar tölurnar hafa komið?
 - Notum rökfylkið (*boolean array*) `found`, þannig að `found[i]` er `true` ef tala `i` hefur komið





Forrit fyrir safnaraverkefni

([Java Visualizer](#))

```
public class CouponCollector {
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        int cardcnt = 0;    // number of cards collected
        int valcnt = 0;     // number of distinct cards

        boolean[] found = new boolean[N];
        while (valcnt < N) {
            int val = (int) (Math.random() * N);
            cardcnt++;
            if (!found[val]) {
                valcnt++;
                found[val] = true;
            }
        }

        System.out.println(cardcnt);
    }
}
```

Fylkið er allt upphafs-
stílt sem *false*

Á meðan fjöldi ólíkra
fundinna spila er $< N$

Ef nýja spilið hefur
ekki sést fyrr þá
merkja við það

Hvað þurfti að fá
mörg spil alls?



Stærðfræðileg greining

Það eru N ólíkar gerðir af spilum, hvað þarftu að safna mörgum þar til þú hefur a.m.k. eitt spil af hverri gerð?

Setning: Væntur fjöldi spila er $N(\frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{N}) = NH_N$

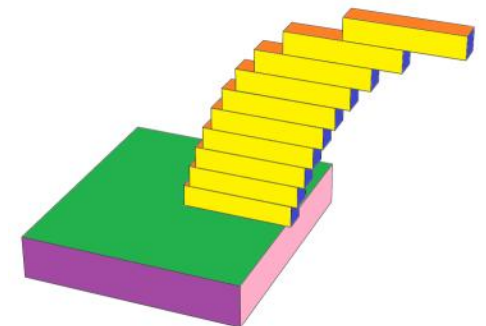
N-ta þýðtalan
(harmonic number)

Þetta gildi er $\approx N \ln N$

Fyrir 52 spil er væntur fjöldi

$$52 * H_{52} = 52 * 4.538 \approx 236$$

H_n er mesta fjarlægð sem stafli af n spilum getur náð fram af borðbrún án þess að detta





Fyrirlestraræfing

4. Hvað prentast út hér að neðan?

```
int[] a = new int[10];  
int[] b;  
b[0] = 2;  
System.out.println(a[0] + ", " + b.length);
```

5. Þegar spilastokkurinn er búinn til þá er spaðatvistur (2 of clubs) fyrsta spilið. Hvaða spil kemur næst?
6. Ef við höfum N -staka rökfylki (*boolean array*) hvernig athugum við hvort öll stök þess séu sönn (`true`)?



Samantekt

- Í þessum tíma:
 - Fylki í Java
 - Dæmi um notkun fylkja
- Í næsta tíma:
 - Tvívíð fylki í Java
 - Dæmi um notkun

Kafli 1.4

Kafli 1.4



HÁSKÓLI ÍSLANDS

ÍÐNADARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD