

HEIMADÆMI 3

TÖL203G Tölvunarfræði 2

Kári Hlynsson¹

Háskóli Íslands

9. febrúar 2023

Verkefni 1

- (a) Bætið við klasann `Card` úr æfingadæminu aðferðinni `toString()`, sem skilar streng með gildi spilsins sem hægt er að prenta út. Þið getið notað ensku upphafsstafina fyrir spaða (S), hjarta (H), tígul (D) og lauf (C). Sömuleiðis fyrir mannsþilin: ás (A), kóngur (K), drottning (Q) og gosi (J). Þannig á aðferðin að skila „H-K” fyrir hjartakóng, „C-5” fyrir laufafimmu o.s.frv.
- (b) Skriðið forritið `CardDeal`, sem tekur á skipanalínunni töluna k sem er á bilinu 1 til 52. Forritið prentar þá út k spil sem valin eru af handahófi úr spilastokki. Til þess að við prentum ekki sömu spilin út aftur, þá er best að búa til 52-spila fylki. Það er fyllt af öllum mögulegum spilum í venjulegri röð og þetta fylki er síðan stokkað. Til þess getið þið notað aðferðina `shuffle` úr `StdRandom`. Síðan prentar forritið út k fyrstu stökin í fylkinu. Skilið kóðanum fyrir forritið og skjáskoti af keyrslu.

Lausn

Hluti (a)

Útfærsluna má sjá í FORRITI 1 fyrir neðan.

```
21 public String toString() {
22     char[] suits = {'S', 'H', 'D', 'C'};
23     String[] ranks = {"A", "2", "3", "4", "5", "6", "7",
24                       "8", "9", "10", "J", "Q", "K"};
25     return suits[suit] + "-" + ranks[rank - 1];
26 }
```

FORRIT 1: Útfærsla á `toString()` í `Card` klasanum

¹Slóð á Github kóða: <https://github.com/lvthnn/TOL203G/tree/master/HD4>

Hluti (b)

```
1 import edu.princeton.cs.algs4.Insertion;
2 import edu.princeton.cs.algs4.StdRandom;
3
4 public class CardDeal {
5
6     /**
7      * Generates sorted array of playing cards.
8      *
9      * @return a deck of playing cards
10     */
11     public static Card[] generate_deck() {
12         Card[] deck = new Card[52];
13         for (int i = 0; i < 52; i++)
14             deck[i] = new Card(i % 4, i % 13 + 1);
15         Insertion.sort(deck);
16
17         return deck;
18     }
19
20     /**
21      * Gathers uniform k-sized sample from deck
22      *
23      * @param k number of cards in selection
24      * @return a random selection of k cards from the deck
25     */
26     public static Card[] deal_cards(int k, Card[] deck) {
27         Card[] sample = new Card[k];
28         StdRandom.shuffle(deck);
29
30         for (int i = 0; i < k; i++) sample[i] = deck[i];
31
32         return sample;
33     }
34
35     public static void main(String[] args) {
36         int num_cards = Integer.parseInt(args[0]);
37         Card[] deck = generate_deck();
38
39         StdRandom.shuffle(deck);
40         Card[] sample = deal_cards(num_cards, deck);
41         for (Card card : sample) System.out.println(card.toString());
42
43     }
44 }
45 }
```

```
*[master] [~/Github/TOL203G/HD4/src/V1]$ java CardDeal 3
C-2
C-4
H-8
*[master] [~/Github/TOL203G/HD4/src/V1]$ java CardDeal 7
C-J
D-3
S-5
S-Q
C-6
D-9
D-8
*[master] [~/Github/TOL203G/HD4/src/V1]$ java CardDeal 15
C-J
D-Q
D-6
S-3
H-9
C-3
S-6
C-10
C-Q
S-10
H-J
C-7
C-6
C-K
C-4
*[master] [~/Github/TOL203G/HD4/src/V1]$ █

[0] 0:nvim- 1:zsh* "Karis-MacBook-Air-2.l" 16:33 09-Feb-23
```

MYND 1: Keyrsla á CardDeal í skel fyrir inntök 3, 7, 15

Verkefni 2

Í útfærslunni á Valröðun í bókinni þá er ekki athugað hvort við þurfum að víxla á stökum `a[i]` og `a[min]`. Ef `i` er jafnt og `min` þá er þessi víxlun óþörf. Bætið við `if`-setningu á undan víxlunarskipuninni í röðunarfallinu `sort` sem athugar hvort það þurfi að víxla. Bætið tímamælingarkóða við báðar útgáfurnar og keyrið þær svo á skránni `32Kints.txt` og athugið hvort það sé einhver hraðamunur. Þið ættuð að keyra hvora útgáfu a.m.k. þrisvar sinnum og taka meðaltalið, því það er alltaf einhver breytileiki í keyrslutímanum. Skilið breytta fallinu og niðurstöðum tímamælinganna.

Lausn

Breytta fallið er fyrir neðan í FORRITI 3.

```
13     public static void sort(Comparable[] a) {
14         int n = a.length;
15         for (int i = 0; i < n; i++) {
16             int min = i;
17             for (int j = i+1; j < n; j++) {
18                 if (less(a[j], a[min])) min = j;
19             }
20             if (min != i) exch(a, i, min);
21             assert isSorted(a, 0, i);
22         }
23         assert isSorted(a);
24     }
```

FORRIT 3: (Hraðvirkari) útfærsla á valröðun

Við höldum upprunalegu útfærslunni í klasanum sem `sort_original()` og útfærum eftirfarandi fall fyrir tímakeyrslu (sjá FORRIT 4).

```
122     public static double timeFunc(Integer[] a, String f) {
123         Stopwatch timer = new Stopwatch();
124         if (f.equals("original")) Selection.sort_original(a);
125         else if (f.equals("fast")) Selection.sort(a);
126         else return 0.0;
127         return timer.elapsedTime();
128     }
```

FORRIT 4: Útfærsla á tímamælingarfalli

Hvað keyrslutíma varðar er munurinn nánast enginn. Meðaltal fyrir útfærsluna okkar keyrt á skránni `32Kints.txt` var 0.6063 sekúndur með staðalfrávik 0.0511 sekúndur. Aftur á móti var meðaltal fyrir upprunalegu útfærsluna 0.5948 sekúndur með staðalfrávik 0.0556 sekúndur. Það getur ekki talist marktækur munur á hraða útfærslanna ($p = 0.3988$) og sá sem má sjá í þessum mælingum er eflaust tilviljanakenndur.

Verkefni 3

Þetta er spurning um hegðun valröðunar og innsetningarröðunar á tilteknu N -staka inntaksfylki:

- (a) Öll stökin í fylkinu hafa sama gildi:
- (i) Hversu marga samanburði notar valröðun?
 - (ii) Hversu marga samanburði notar innsetningarröðun?
- (b) Fylkið er óraðað en inniheldur aðeins tvö ólík gildi, A og B . Fjöldi staka af hvoru gildi er óþekktur:
- (i) Hversu marga samanburði notar valröðun?
 - (ii) Hversu marga samanburði notar innsetningarröðun?

Lausn

Hluti (a)

- (i) Valröðun notar alltaf sama fjölda samanburða því það ítrar frá vísinum i og út í enda fylkisins. Fjöldi samanburða er því sem áður $(N-1) + (N-2) + \dots + 2 + 1 \sim N^2/2$.
- (ii) Innsetningarröðun stöðvar alltaf ef $a[j] \leq a[j-1]$, svo í þessu tilviki myndum við alltaf bara líta eitt stak aftur fyrir okkur í röðun á fylkinu. Því er heildarfjöldi samanburða $N-1 \sim N$.

Hluti (b)

- (i) Eins og áður sagði er valröðun ekki háð inntaki og því höfum við aftur fjölda samanburða $(N-1) + (N-2) + \dots + 2 + 1 \sim N^2/2$.
- (ii) Látum n og m tákna fjöldann af gildunum A og B í fylkinu, í þeirri röð. Þá er $N = n + m$ heildarföldi staka í fylkinu. Án skerðingar á víðgildni athugum við að ef $n \gg m$ þá erum við nokkurn veginn að fást við (ii) í (a)-lið. Ef svo er ekki og $a \approx m$ þá er fjöldi samanburða áður en stak finnur sinn stað í raðaða fylkinu $\frac{1}{2}(N-i)$. Við fáum því að heildarfjöldi samanburða er $\frac{1}{2}(N-1) + \frac{1}{2}(N-2) + \dots + 1 + \frac{1}{2} \sim N^2/4$.

Verkefni 4

Við getum skilgreint röðunarreikniritið *Slembiröðun*, sem virkar þannig að á meðan fylkið er ekki raðað þá veljum við tvo vísa i og h af handahófi (á milli 0 og $N - 1$). Ef stök $a[i]$ og $a[j]$ eru í langri röð í fylkinu þá víxlum við á þeim og höldum áfram. Forritið þetta reiknir í Java (þið getið notað `Selection.java` sem fyrirmynd). Takið tímann á keyrslu á `1Kints.txt`. Keyrið forritið ykkar a.m.k. 5 sinnum og skoðið breytileikann á tímanum. Skilið Java kóðanum fyrir fallið og tímunum á keyrslunum.

Lausn

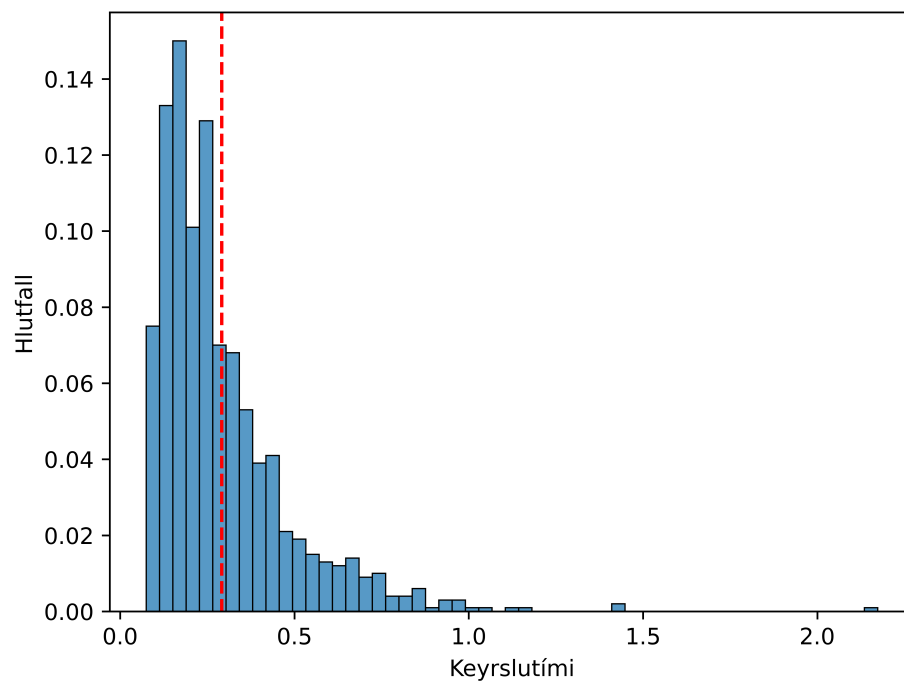
Fallið er fyrir neðan í FORRITI 5.

```
34 public static void sort(Comparable[] a) {  
35     int n = a.length;  
36  
37     while (!isSorted(a)) {  
38         int i, j;  
39         i = StdRandom.uniformInt(0, n);  
40         j = StdRandom.uniformInt(i, n);  
41         if (less(a[j], a[i])) exch(a, i, j);  
42     }  
43 }
```

FORRIT 5: Útfærsla á slembiröðun

Forritið var keyrt 1000 sinnum og úttakinu beint í textaskrá. Meðaltal var 0.2912 sekúndur með staðalfráviki 0.1932 sekúndur sem er talsvert hærra en það sem við höfum séð hingað til. Mynd af dreifingu á keyrslutíma er á næstu síðu á MYND 2 ásamt töflu sem sýnir keyrslutímana.

0.4 0.12 0.132 0.193 0.153 0.211 0.368 0.114 0.242 0.273 0.147 0.28 0.195
0.262 0.378 0.155 0.275 0.141 0.145 0.244 0.52 0.222 0.104 0.165 0.585 0.967
0.741 0.292 0.167 0.257 0.115 0.335 0.241 0.565 0.272 0.165 0.311 0.17 0.27
0.18 0.512 0.124 0.301 0.179 0.258 0.262 0.157 0.128 0.412 0.264 0.317 0.168
0.087 0.218 0.133 0.285 0.158 0.135 0.236 0.152 0.63 0.382 0.135 0.166 0.159
0.352 0.405 0.239 0.337 0.193 0.148 1.136 0.947 0.725 0.428 0.142 0.255 0.197
0.368 0.263 0.212 0.33 0.254 0.43 0.341 0.248 0.165 0.2 0.203 0.521 0.236 0.523
0.249 0.296 0.213 0.727 0.172 0.238 0.338 0.243 0.112 0.388 0.175 0.208 0.23
0.259 0.182 0.407 0.314 0.538 0.114 0.649 0.321 0.29 0.406 0.162 0.355 0.23



MYND 2: Dreifing á keyrslutíma slembiröðunar fyrir $N = 1000$ ($n = 1000$)

0.541 0.356 0.165 0.254 0.098 0.128 0.293 0.315 0.191 0.123 0.338 0.458 0.197
0.611 0.147 0.663 0.776 0.684 0.39 2.173 1.051 0.561 0.368 0.398 0.112 0.231
0.134 0.488 0.602 0.261 0.247 0.346 0.181 0.161 0.174 0.277 0.38 0.381 0.243
0.584 0.113 0.215 0.266 0.183 0.839 0.147 0.27 0.784 0.188 0.172 0.232 0.11
0.236 0.401 0.17 0.244 0.094 0.223 0.205 0.15 0.117 0.237 0.342 0.147 0.228
0.11 0.199 0.403 0.226 0.568 0.289 0.558 0.197 0.381 0.442 0.233 0.148 0.587
0.093 0.099 0.199 0.218 0.324 0.105 0.848 0.377 0.139 0.501 0.209 0.37 0.16
0.352 0.261 0.489 0.165 0.326 0.181 0.116 0.236 0.228 1.433 0.143 0.116 0.172
0.263 0.128 0.209 0.09 0.177 0.499 0.159 0.254 0.112 0.35 0.434 0.635 0.248
0.289 0.262 0.292 0.254 0.202 0.096 0.843 0.164 0.298 0.195 0.096 0.121 0.259
0.228 0.653 0.685 0.309 0.229 0.29 0.853 0.588 0.327 0.196 0.128 0.369 0.133
0.156 0.291 0.291 0.147 0.262 0.197 0.266 0.607 0.183 0.518 0.273 0.127 0.157
0.182 0.164 0.341 0.246 0.242 0.316 0.454 0.12 0.673 0.661 0.116 0.225 0.544
0.456 0.254 0.275 0.266 0.525 0.095 0.536 0.136 0.737 0.255 0.464 0.386 0.347
0.148 0.19 0.209 0.338 0.17 0.23 0.128 0.186 0.161 0.261 0.423 0.368 0.268 0.623
0.337 0.213 0.098 0.466 0.157 0.19 0.276 0.379 0.555 0.099 0.185 0.357 0.335
0.164 0.171 0.21 0.346 0.377 0.631 0.233 0.385 0.281 0.164 0.277 0.23 0.298
0.097 0.349 0.31 0.432 0.544 0.18 0.353 0.398 0.138 0.431 0.303 0.479 0.154
0.114 0.222 0.238 0.128 0.093 0.339 0.255 0.383 0.143 0.253 0.332 0.347 0.25
0.184 0.23 0.331 0.204 0.124 0.375 0.315 0.801 0.162 0.508 0.267 0.214 0.11
0.39 0.22 0.732 0.505 0.185 0.226 0.294 0.182 0.254 0.252 0.255 0.654 0.388
0.221 0.152 0.576 0.479 0.452 0.18 0.117 0.372 0.264 0.976 0.702 0.503 0.265
0.322 0.407 0.157 0.248 0.347 0.19 0.637 0.116 0.37 0.727 0.707 0.116 0.185
0.206 0.142 0.809 0.083 0.431 0.094 0.083 0.133 0.204 0.354 0.633 0.296 0.362
0.094 0.194 0.159 0.14 0.372 0.153 0.167 0.255 0.407 0.242 0.314 0.151 0.174
0.299 0.147 0.165 0.536 0.503 0.172 0.178 0.226 0.413 0.159 0.312 0.257 0.168
0.167 0.376 0.513 0.148 0.104 0.178 0.145 0.327 0.191 0.163 0.222 0.183 0.187
0.19 0.107 0.169 0.448 0.197 0.234 0.47 0.26 0.26 0.157 0.106 0.423 0.309 0.254
0.45 0.28 0.165 0.246 0.222 0.317 0.12 0.406 0.297 0.144 0.402 0.213 0.13 0.142
0.254 0.392 0.143 0.147 0.203 0.101 0.128 0.248 0.392 0.26 0.208 0.124 0.312
0.181 0.24 0.245 0.253 0.103 0.193 0.159 0.137 0.183 0.234 0.194 0.201 0.282
0.3 0.121 0.223 0.197 0.159 0.498 0.191 0.134 0.227 0.52 0.302 0.432 0.141
0.381 0.438 0.106 0.455 0.116 0.476 0.182 0.113 0.295 0.452 0.295 0.31 0.456
0.303 0.111 0.102 0.12 0.246 0.13 0.27 0.435 0.325 0.27 0.146 0.276 0.2 0.212
0.154 0.289 0.164 0.53 0.31 0.455 0.21 0.158 0.222 0.106 0.166 0.643 0.232 0.175
0.261 0.423 0.324 0.219 0.124 0.296 0.26 0.245 0.138 0.243 0.459 0.178 0.426
0.199 0.259 0.456 0.108 0.488 0.317 0.811 0.36 0.235 0.184 0.389 0.43 0.325
0.248 0.107 0.189 0.213 0.098 0.257 0.387 0.288 0.351 0.213 0.241 0.571 0.241
0.126 0.424 0.431 0.863 0.214 0.565 0.427 0.163 0.578 0.442 0.101 0.328 0.331
0.205 0.707 0.127 0.178 0.676 0.667 0.737 0.325 0.188 0.168 0.122 0.668 0.32
0.26 0.348 0.798 0.077 0.339 0.55 0.38 0.246 0.11 0.44 0.118 0.397 0.208 0.439

0.219 0.937 0.238 0.24 0.347 0.307 0.282 0.101 0.167 0.319 0.159 0.139 0.407
0.174 0.426 0.369 0.165 0.289 0.17 0.141 0.126 0.272 0.33 0.21 0.315 0.309 0.28
0.308 0.289 0.198 0.156 0.151 0.102 0.115 0.144 0.426 0.449 0.153 0.238 0.155
0.142 0.286 0.76 0.1 0.401 0.25 0.168 0.075 0.195 0.128 0.161 0.281 0.338 0.409
0.527 0.193 0.151 0.284 0.286 0.21 0.185 0.13 0.111 0.089 0.154 0.096 1.151
0.126 0.134 0.338 0.241 0.207 0.167 0.157 0.466 0.407 0.265 0.365 0.129 0.116
0.087 0.368 0.526 0.119 0.7 0.113 0.185 0.255 0.242 0.35 0.379 0.336 0.2 0.142
0.694 0.101 0.67 0.157 0.771 0.115 0.283 0.357 0.58 0.335 0.889 0.211 0.376
0.125 0.179 0.098 0.479 0.091 0.129 0.168 0.093 0.32 0.115 0.149 0.393 0.704
0.458 0.25 0.17 0.235 0.306 0.196 0.664 0.135 1.436 0.127 0.259 0.24 0.234
0.255 0.179 0.176 0.572 0.346 0.293 0.121 0.168 0.125 0.248 0.235 0.217 0.259
0.319 0.106 0.222 0.597 0.458 0.332 0.35 0.152 0.127 0.228 0.12 0.208 0.11
0.188 0.217 0.178 0.145 0.131 0.166 0.929 0.161 0.354 0.172 0.132 0.308 0.104
0.256 0.816 0.353 0.399 0.395 0.086 0.313 0.142 0.635 0.164 0.709 0.162 0.129
0.282 0.184 0.751 0.309 0.196 0.168 0.21 0.142 0.244 0.109 0.43 0.618 0.415
0.24 0.248 0.162 0.467 0.425 0.424 0.107 0.289 0.165 0.154 0.158 0.094 0.278
0.419 0.134 0.209 0.705 0.108 0.125 0.191 0.152 0.676 0.351 0.347 0.247 0.196
0.189 0.512 0.165 0.249 0.12 0.164 0.645 0.214 0.133 0.42 0.607 0.103 0.155
0.222 0.443 0.139 0.993 0.08 0.24 0.181 0.189 0.129 0.091 0.183 0.307 0.291
0.217 0.141 0.114 0.242 0.138 0.277 0.429 0.26 0.251 0.156 0.34 0.099 0.163
0.222 0.465 0.187 0.286 0.316 0.142 0.219 0.38 0.163 0.139 0.234 0.286 0.278
0.099 0.747 0.22 0.169 0.121 0.22 0.341 0.185 0.115 0.489 0.134 0.291 0.338
0.161 0.138 0.104 0.245 0.857 0.169 0.156 0.208 0.299 0.195 0.232 0.467 0.172
0.345 0.108 0.559 0.123 0.462 0.177 0.12 0.354 0.239 0.643 0.104 0.561 0.12
0.093 0.101 0.173 0.695 0.306 0.249 0.194 0.398 0.334 0.123 0.235 0.174 0.343
0.235 0.142 0.194 0.245 0.096 0.119 0.181 0.141 0.961 0.142 0.131 0.465 0.157
0.076 0.236

Verkefni 5

Nota á Shell röðun með $3x + 1$ skrefstærðum á 10-staka fylki. Þá eru tvær skrefstærðir: 1 og 4 (reyndar er byrjað með $h = 4$).

- (a) Hvert er besta inntak fyrir þessa tegund af Shellröðun? Rökstyðjið og sýnið heildarfjölda samanburða fyrir 10-staka fylki.
- (b) Sýnið hvernig þessi Shell röðun virkar á 10-staka fylki í öfugri röð (t.d. 10, 9, ..., 2, 1). Sýnið fylkið eftir hvora umferð og fjölda samanburða. Berið fjölda samanburða hér saman við fjölda samanburða sem innsetningarröðun myndi nota á þessu fylki.

Lausn

Hluti (a)

Besta tilvikið kemur upp þegar fylkið er fyrirfram raðað og innri lykkjan keyrir ekki. Kostnaðurinn er einfaldlega að kíkja í gegnum allar hlutrunurnar svo kostnaðurinn er $\Theta(N)$.

Hluti (b)

Setjum upp rakningar af hlutröðunum í TÖFLU 2 og TÖFLU 1 hér fyrir neðan.

| | | | | | | | | | | |
|---------|----|-------|-------|---|---|-------|-------|---|----|---|
| $h = 4$ | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| | 10 | ————— | | | 6 | ————— | | | 2 | |
| | | 9 | ————— | | | 5 | ————— | | | 1 |
| | 2 | 1 | 8 | 7 | 6 | 5 | 4 | 3 | 10 | 9 |

TAFLA 1: Fyrsta h -röðunin með $h = 4$.

Í 4-röðuninni (TAFLA 1 er fjöldi samanburða 6 því við röðum tveimur hlutrunum sem kosta hver um sig 3 samanburði. Næst kemur 1-röðunin í TÖFLU 2 fyrir neðan sem er venjuleg innsetningarröðun, en í henni eru framkvæmdir 25 samanburðir. Til að sjá betri útskýringu, athuga TÖFLU 3.

| | | | | | | | | | | |
|---------|---|---|---|---|---|---|---|---|----|----|
| $h = 1$ | 2 | 1 | 8 | 7 | 6 | 5 | 4 | 3 | 10 | 9 |
| | 2 | 1 | 8 | 7 | 6 | 5 | 4 | 3 | 10 | 9 |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

TAFLA 2: Seinni röðunin með $h = 1$.

| | | a[] | | | | | | | | | | Samanburðir, S |
|-----|-----|-----|---|---|---|---|---|---|---|----|----|------------------|
| i | j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| | | 2 | 1 | 8 | 7 | 6 | 5 | 4 | 3 | 10 | 9 | |
| 1 | 0 | 1 | 2 | 8 | 7 | 6 | 5 | 4 | 3 | 10 | 9 | 1 |
| 2 | 2 | 1 | 2 | 8 | 7 | 6 | 5 | 4 | 3 | 10 | 9 | 1 |
| 3 | 2 | 1 | 2 | 7 | 8 | 6 | 5 | 4 | 3 | 10 | 9 | 2 |
| 4 | 2 | 1 | 2 | 6 | 7 | 8 | 5 | 4 | 3 | 10 | 9 | 3 |
| 5 | 2 | 1 | 2 | 5 | 6 | 7 | 8 | 4 | 3 | 10 | 9 | 4 |
| 6 | 2 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 3 | 10 | 9 | 5 |
| 7 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 10 | 9 | 6 |
| 8 | 8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 10 | 9 | 1 |
| 9 | 8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 2 |
| | | | | | | | | | | | | $\sum S = 25$ |

TAFLA 3: Rakning af 1-röðun í Shell röðun með $N = 10$

Prófum nú að bera þetta saman við fjölda samanburða ef við myndum keyra innsetningarröðun á upprunalega fylkið sem gefið var í dæminu:

| | | a[] | | | | | | | | | | Samanburðir, S |
|-----|-----|-----|----|----|----|----|----|----|----|----|----|------------------|
| i | j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| | | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | |
| 1 | 0 | 9 | 10 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 1 |
| 2 | 0 | 8 | 9 | 10 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 2 |
| 3 | 0 | 7 | 8 | 9 | 10 | 6 | 5 | 4 | 3 | 2 | 1 | 3 |
| 4 | 0 | 6 | 7 | 8 | 9 | 10 | 5 | 4 | 3 | 2 | 1 | 4 |
| 5 | 0 | 5 | 6 | 7 | 8 | 9 | 10 | 4 | 3 | 2 | 1 | 5 |
| 6 | 0 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 3 | 2 | 1 | 6 |
| 7 | 0 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 2 | 1 | 7 |
| 8 | 0 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 8 |
| 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 9 |
| | | | | | | | | | | | | $\sum S = 45$ |

TAFLA 4: Innsetningarröðun á fylki í öfugri röð með $N = 10$.

Eins og má sjá er sparnaðurinn talsverður við að nota Shell röðun, en það munar 20 samanburðum á röðunarreikniritunum.