

Heimadæmi02. Tölvunarfræði02

Donn Eunice Bartido deb5@hi.is

Janúar 2023

1. [Spilaklasi]

Þetta dæmi byggir á æfingadæminu úr dæmatímunum.

- 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 1. Forritið prentar þá út k spil sem valin eru af handhófi 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 út StdRandom. Síðan prentar forritið út k fyrstu stökin í fylkinu. Skilið kóðanum fyrir forritið og skjáskoti af keyrslu.

a)

```
In [5]: class Card:
    def __init__(self, suit, value):
        self.suit = suit
        self.value = value

    def toString(self):
        suit_map = {'S': 'Spaði', 'H': 'Hjarta', 'D': 'Tígull', 'C': 'Lauf'}
        value_map = {'A': 'Ás', 'K': 'Kóngur', 'Q': 'Drolla', 'J': 'Gosi'}

        suit_str = suit_map.get(self.suit, self.suit)
        value_str = value_map.get(self.value, self.value)

        return f"{suit_str}-{value_str}"
```

b)

```
In [6]: class CardDeal:
    def __init__(self, num_cards):
        self.num_cards = num_cards
        self.deck = []
        suits = ['S', 'H', 'D', 'C']
        values = ['A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K']
        for suit in suits:
            for value in values:
                self.deck.append(Card(suit, value))
```

```

def deal(self):
    random.shuffle(self.deck)
    return self.deck[:self.num_cards]

if __name__ == '__main__':
    num_cards = int(input("Enter the number of cards to deal: "))
    card_deal = CardDeal(num_cards)
    cards = card_deal.deal()

    for card in cards:
        print(card.toString())

```

Tígull-8

Lauf-10

Spaði-8

2

In [26]:

```

def sort(a):
    n = len(a)
    for i in range(n):
        min_idx = i
        for j in range(i + 1, n):
            if a[j] < a[min_idx]:
                min_idx = j
        if a[i] != a[min_idx]:
            a[i], a[min_idx] = a[min_idx], a[i]

```

Tímamæling án auka if setningar:

- Fyrsta keyrsla: 0.795
- Önnur keyrsla: 0.96
- Þriðja keyrsla: 0.811
- Fjórða keyrsla: 0.883

Tímamæling með auka if setningu:

- Fyrsta keyrsla: 0.794
- Önnur keyrsla: 0.806
- Þriðja keyrsla: 0.802
- Fjórða keyrsla: 0.879

3

a)

1. Í valröðun er fyrsta stakið borið saman við öll hin til að finna minnsta, sem eru $N-1$ samanburðir. Síðan er næsta stak borið saman við restina og svo koll af kolli. Á endanum verða samanburðirnir $N-1 + N-2 + N-3 + \dots + 1 = N*(N-1)/2$
2. Ef öll stök eru jöfn fer innsetningarröðun einu sinni í gegnum alla ytri for loop-una til að bera saman stakið sem er valið við næsta stak en aldrei í gegnum innri loop-una.

Það eru $N-1$ samanburðir.

b)

1. Valröðun mun alltaf hafa $N*(N-1)/2$ samanburði því hún fer alltaf í gegnum alla loopuna.
2. Innsetningarröðun mun gera $|B|-1 + |A|*|B|$ samanburði. Gerum ráð fyrir að sé verið að raða í stafrófsröð, þá ber hún fyrst öll B stökin saman þangað til hún kemur að fyrsta A stakinu og svo rekur hún sig til baka og ber A stakið saman við öll B stökin og eins með restina af A stökunum.

4)

[Röðun] Við getum skilgreint röðunarreikniritið Slembiröðun, sem virkar þannig að á meðan fylkið er ekki raðað þá veljum við tvo vísa i og j af handahófi (á milli 0 og $N-1$). Ef stök $a[i]$ og $a[j]$ eru í rangri röð í fylkinu þá víxlum við á þeim og höldum áfram. Forritið þetta reiknirit 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 í tímanum. Skilið Java kóðanum fyrir fallið og tímunum á keyrslunum.

5.

[Shell röðun] 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]).
- b. Sýnið fylkið eftir hvora umferð og fjölda samanburða. Berið fjölda samanburða hér saman við fjölda samanburða sem Innsetningarröðun myndi nota á þessu fylki.

a)

Besta inntakið væri í réttri röð. Þá þarf forritið að gera 22 samanburði.

In []: