

Heimadæmi 1, TÖL 203, Elfar Oliver Sigurðarson, eos35@hi.is

1a. Sjálfur nota ég aðallega aðferð 4 en ég highlighta ekkert nú til dags, tek bara screengrab. Nota einnig 7 til að reyna festa eitthvað í minninu þar sem einn lestur er aldrei nóg. 8 er hentugast í námsefnum eins og stærðfræði eða forritun þar sem lærdómur er mjög mikið í að endurtaka eitthvað physically.

1b. Þar stendur að 8 og 9 eru bestu aðferðirnar, sem ég er sammála. Þá er aðferð 8, sem er practice testing(endurtekning), endurtekið eftir ákveðnu skipulagi og þannig festist allt mjög svo líklega betur í hausnum á þeim sem er að læra. Ég er ekki ósammála að það sem flokkast sem low utility sé low utility, en aðferð 4(highlighting) gagnast t.d. mjög mikið til þess að upprifja námsefni ef það er rétt nýtt. T.d. fer nemandi yfir heilan kafla og skilur algjöran meirihluta en highlightar það sem var aðeins að vefjast fyrir þeim og getur haft það á hreinu seinna meir

2. Virkar stofur og virkir fyrirlestrar sem innihalda einhver verkefni enda í hærri einkunnir sem lokaútkomu áfanga eða prófa frekar en stofur og fyrirlestrar sem þylja bara upp námsefnið og segja svo að tíminn sé búinn. Eina ferðina enn hefur verið sannað að fólk lærir með því að framkvæma aftur og aftur frekar en að *bara* hlusta og labba svo út

3.

```
PS C:\Nám\Vor25\TÖL203\Heimadæmi\Heimadæmi1> java Runs 3
4 3 2 3 2 1 2 1 0
Number of flips: 9
PS C:\Nám\Vor25\TÖL203\Heimadæmi\Heimadæmi1> |
```

4. Dæmið er:

- a. 4 3 5 2 7 6 8 1 9
- b. 2 4 6 8 7 5 1 3 9
- c. 1 2 3 7 8 6 9 5 4

- a) 1, 2, 3, 4-, 3-, 5, 5-, 2-, 6, 7, 7-, 6-, 8, 8-, 1-, 9, 9-
- b) 1, 2, 2-, 3, 4, 4-, 5, 6, 6-, 7, 8, 8-, 7-, 5-, 1-, 3-, 9, 9- er best I can do en mér skilst ég geti ekki pop-að ef það er annað stak fyrir sem gerist í 1 og 3 næstum í lokin
- c) 1-, 2-, 3-, 4, 5, 6, 7-, 8-, 6-, 9---

5. Fyrir neðan er

- a) kóðinn minn fyrir addBack
- b) terminal fyrir og eftir breytingu á main sem sýnir muninn á Back og Front
- c) teikning sem sýnir hvernig a b c d fer inní listann

```

public void addBack(String item) {
    if (isEmpty()) {
        first = new Node();
        first.item = item;
    } else {
        Node current = first;
        while (current.next != null) { // Fer í last node
            current = current.next;
        }
        current.next = new Node(); // Ný node í endann
        current.next.item = item; // Assigna item í ný node
    }
    N++;
}

```

```

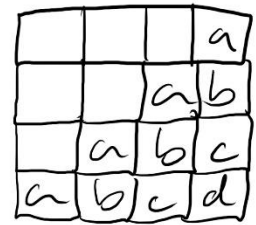
Run | Debug
86 public static void main(String[] args) {
87     LinkedListOfStrings list = new LinkedListOfStrings();
88     for (int i=0; i < args.length; i++) {
89         list.addBack(args[i]);
90     }
91
92     System.out.println("Number of items: " + list.size());
93
94     list.printList();
95
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Number of flips: 9
PS C:\Nám\Vor25\TÖL203\Heimadæmi\Heimadæmi>
* History restored

PS C:\Nám\Vor25\TÖL203\Heimadæmi\Heimadæmi> javac LinkedListOfStrings.java
PS C:\Nám\Vor25\TÖL203\Heimadæmi\Heimadæmi> java LinkedListOfStrings a b c d
Number of items: 4
d->c->b->a->
PS C:\Nám\Vor25\TÖL203\Heimadæmi\Heimadæmi> javac LinkedListOfStrings.java
PS C:\Nám\Vor25\TÖL203\Heimadæmi\Heimadæmi> java LinkedListOfStrings a b c d
Number of items: 4
a->b->c->d->
PS C:\Nám\Vor25\TÖL203\Heimadæmi\Heimadæmi>

```

p v addBack



p v addFront

