

HEIMADÆMI 2

TÖL203G Tölvunarfræði 2

Kári Hlynsson

Háskóli Íslands

Vormisseri 2023

Verkefni 1.

Lesið þessa grein úr tímaritinu Science. Segið frá meginniðurstöðum hennar í örfáum orðum.

Úrlausn Í greininni eru yfirburðir endurheimtunar (*memory retrieval*) sem námsaðferð undirstrikaðir. Að lesa texta aftur til þess að festa efni betur í minni reynist síður skilvirkt, en talið er að það myndi ekki jafn sterkar minnisbrautir í heilanum. Jafnframt hefur endurheimt endurtekið reynst vera sterkasta aðferðin við að leggja námsefni á minnið borið saman við fjölda annarra aðferða.

Rannsóknir sýna að þegar streita hefur neikvæð áhrif á getu einstaklings til að framkalla atriði úr minni. Þetta er talið vera vegna þess að stresshormónið cortisol hefur áhrif á þau svæði í heilanum sem koma að minnisendurheimt, þ.á m. dreka (*hippocampus*). Hormónið binst efnaviðtökum á þessum svæðum og veldum röskunum í ferlum sem koma að minnisendurheimt. Hin umrædda aðferð er öflug í ljósi þess að hún myndar nýjar brautir fyrir upplýsingar að ferðast þrátt fyrir raskanirnar, og þannig hefur streita ekki jafn mikil áhrif og ella.

Verkefni 2.

Þessi grein fjallar um tilraun með virka (*active*) kennsluaðferð í eðlisfræði í Harvard háskóla. Lesið a.m.k. inngang greinarinnar og skoðið súluritin í Myndum 1 og 2 (*Fig. 1* og *Fig. 2*). Í þessum súluritum koma fyrir hugtökin TOF (*Test of Learning*) og FOL (*Feeling of Learning*). Skrifðið nokkrar setningar þar sem þið reynið að útskýra hvers vegna fyrstu tvær súlurnar (með skyggðum bakgrunni) í myndunum eru svona ólíkar hinum súlunum.

Úrlausn Í þessari rannsókn vildu rannsóknarmenn sýna kosti þess að kennsla færi fram á virku formi — þ.e. að námi væri miðað í gegnum samvinnu nemanda, hópverkefnum, umræðuhópum og almennt aðferðum sem svipa ekki til hefðbundinna fyrirlestra. Svo virðist vera sem virk kennsla sýni mikla yfirburði í því hvernig efnið kemst til skila ef marka má frammistöðu nemenda á prófum, en þessi aðferð virðist aftur á móti síður vekja tilfinningu um að skilja efnið vel. Að því leyti er algengara að nemendur upplifi ósætti með kennslu og því er talið að hið hefðbundna fyrirkomulag fyrirlestra er fast í sessi.

Ef litið er á gröfin má sjá einkunnir úr prófi sem mat þekkingu á námsefninu, en til hliðsjónar er spurningalisti sem metur tilfinningu nemanda fyrir náminu. Sjá má að nemendur sem fengu virka kennslu voru með áberandi hærri einkunn á prófinu, en virðast allt í allt vera ósáttari með kennsluhætti. Þetta er sýnt á myndum 1 og 2 í greininni en þetta er gert fyrir tvo mismunandi áfanga.

Verkefni 3.

Hægt er að hugsa sér að hlaði sé notaður til þess að útfæra bak-örina í vafra. Þegar notandi fer á nýja vefsíðu þá er núverandi síða sett efst á hlaða (*push*). Þegar notandinn ýtir svo á bakörina þá er efsta síðan á hlaðanum tekin (*pop*) og sýnd.

- (a) Vafrar hafa líka áfram-ör, sem er virk ef notandinn hefur farið til baka. Hvernig væri hægt að útfæra hana með aðstoð hlaða?
- (b) Mörg ritvinnsluforrit hafa aðgerðirnar *undo/redo* (afturkalla/endurgera). Hvernig væri hægt að útfæra þær með hlöðum?

Úrlausn (a) Við notum tvo hlaða, N og P . Hlaðinn P útfærir bak-örina en í hvert skipti sem við ýtum á bak-örina þá grípum við skil *pop*-aðgerðarinnar á P og setjum á N . Ef við ýtum á áfram takkann er efsta stakið tekið af N (*pop*) og sýnt.

- (b) Útfærslan er í raun sú sama og í (a) lið. Við höfum tvo hlaða, U og R . Í hvert skipti sem forritið nemur breytingar í skjalinu er upplýsingum um breytingarnar bætt ofan á U . Að smella á *Undo* hnappinn sækir efsta stakið af U en skilunum er bætt á R .

Ef við ýtum síðan á *Redo* sækjum við efsta stakið (aðgerðina sem var afturkölluð) af R og færum aftur inn í skjalið. Hvert skipti sem ritvinnsluforritið nemur breytingar í skjalinu tæmum við R , því þá er ekki hægt að endurgera breytingar án þess að yfirskrifa skjalið. U helst óbreyttur.

Verkefni 4.

Notendaforrit framkvæmir blöndu af *push* og *pop*-aðgerðum. *push*-aðgerðin setur heiltölurnar $0, 1, \dots, 9$ í röð á hlaðann, en *pop*-aðgerðirnar prenta út skilagildið. Sýnið hvenær *pop*-aðgerðirnar koma í röð aðgerða í eftirfarandi úttaksrunum. Fyrir þær úttaksrunur sem ekki eru mögulegar og útskýrið hvers vegna.

(a) 3 2 4 6 7 5 1 9 8 0

(b) 1 2 4 0 3 5 6 8 9 7

(c) 1 2 0 4 5 3 7 8 6 9

Úrlausn (a) 0 1 2 3 - - 4 - 5 6 - 7 - - - 8 9 - - -

(b) Það er ekki hægt að mynda þessa runu með hlaða. Við getum kallað fram fyrstu þrjá stafina með 0 1 - 2 - 3 4 - en þá eru stökin eftir á hlaðanum 3 og 0, með hið síðurnefnda neðst á hlaðanum. Því verður 3 að koma á undan 0 svo það er ekki hægt að mynda þessa röðun.

(c) 0 1 - 2 - - 3 4 - 5 - - 6 7 - 8 - - 9 -

Verkefni 5.

Notið tengda listann fyrir strengi: `LinkedListOfString.java` og bætið við hann aðferðinni `delBack()` sem eyðir út aftasta hnútinum og skilar strengnum sem hann inniheldur. Athugið að það eru nokkur jaðartilvik sem aðferðin þarf að ráða við. Uppfærið einnig prófunarfallið þannig að það noti aðferðina til að eyða öllum stökum út úr þriggja staka lista (sem það býr til). Sýnið aðferðina og skjámynd af keyrslu.

Úrlausn Forritskóðinn er sýndur hér fyrir neðan.

```
1 // delete and return the last item in the list
2 public String delBack() {
3     if (isEmpty())
4         throw new NoSuchElementException("No items in list");
5
6     Node curr = first;
7     String item = first.item;
8
9     if (N == 1) {
10        curr = null;
11    } else {
12        while (curr.next.next != null)
13            curr = curr.next;
14
15        item = curr.next.item;
16        curr.next = null;
17    }
18
19    N--;
20    return item;
21 }
22
```

Útfærum síðan main fall til að prófa þessa aðferð:

```
1 // test client
2 public static void main(String[] args) {
3     LinkedListOfStrings list = new LinkedListOfStrings();
4     list.addFront("item1");
5     list.addFront("item2");
6     list.addFront("item3");
7
8     list.printList();
9     list.delBack();
10 }
```

```

11     list.printList();
12     list.delBack();
13
14     list.printList();
15     list.delBack();
16
17     list.printList();
18 }
19

```

Fyrir neðan er sýnt "skjáskot" af keyrslu, eða hvert úttakið er.

```

1    $ java LinkedListOfStrings
2    item3
3    item2
4    item1
5    item3
6    item2
7    item3
8

```

Sem stemmir við aðgerðaröðina: í fyrstu höfum við `item1`, `item2` og `item3` og prentum. Köllum á `delBack()` í fyrsta sinn en þá eyðist `item1` aftast á listanum og eftir sitja `item3` og `item2`. Þegar við köllum á `delBack()` í annað sinn eyðist `item2` og einungis `item3` er eftir. Þegar við köllum í síðasta skipti á aðferðina er ekkert eftir á listanum og við prentum tóma línu.

Ég setti link á kóðann hér. Það ætti að virka að keyra hann ef sýnidæmið fyrir ofan er ekki fullnægjandi:

<https://github.com/lvthnn/TOL203G/blob/master/HD1/D5/LinkedListOfStrings.java>