Heimadæmi 5, TÖL203. Elfar Oliver Sigurðarson, eos35@hi.is

Æfingadæmi1. * = pop. Poppar alltaf stærstu töluna sem er búið að pusha inn

8 3 2 * 10 7 * 5 12 4 * * * 9 * * 6 * * *

8 10 12 7 5 9 4 6 3 2

Æfingadæmi2.

a. Barn efsta/barnabarn efsta gegnum hitt barnið. Eða öfugt.
b. Þriðja stærsta kemst ekki neðar en að vera barnabarn efsta hnúts
c. Fjórða stærsta kemst í hvaða sæti sem er nema efsta

Heimadæmin

1. [Mergesort] Röðunarreiknirit er sagt vera stöðugt (stable) ef það ruglar ekki innbyrðisröð staka með sama gildi, þ.e. ef a[i] = a[j] með i < j í upphafi röðunar, þá er stakið sem var í a[i] áfram fyrir framan stakið sem var í a[j] eftir röðunina. Í þessu dæmi eigið þið að rökstyðja að Mergesort sé stöðug röðun, eins og hún er útfærð í kennslubókinni í Merge.java

   Rökin úr glæru 6 á fyrirlestri 9 er:

```
int i = lo, j = mid+1;
for (int k = lo; k <= hi; k++)
{
   if      (i > mid)              a[k] = aux[j++];
   else if (j > hi)               a[k] = aux[i++];
   else if (less(aux[j], aux[i])) a[k] = aux[j++];
   else                           a[k] = aux[i++];
}
```

   Ef vinstri hluti tómur þá taka úr hægri hluta
   Ef hægri hluti tómur þá taka úr vinstri hluta
   Annars bera saman fremstu stökin

   Hér er fyrst skoðað hvort helmingarnir séu tómir eða ekki, annars eru fremstu stökin borin saman. else if (less(auk[j], aux[i])) sem ber saman fremstu stökin myndi setja stakið úr j fyrst og i næst ef stökin eru jafnstór.

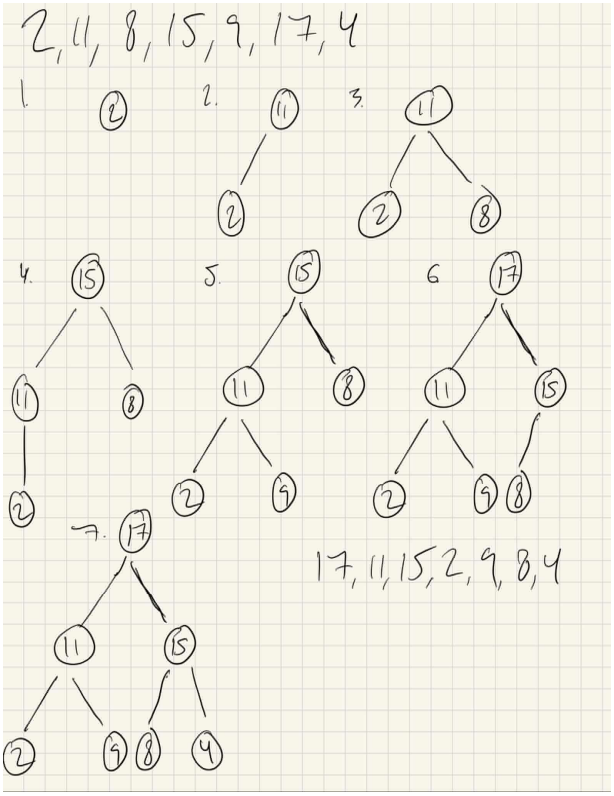
2. Ég breytti sort ekki neitt, því ég fékk alltaf error og fékk það ekki til að virka. Breytti bara main og fékk CUTOFF útprentað fyrir öll 25. CUTOFFið er bara for loop inní main fallinu(sem mér skilst leiðbeiningarnar biðja um). Fylkið er búið til inní forinu þar sem forið fer frá 1 uppí 25.

```java
public static void main(String[] args) {
    String[] a = StdIn.readAllStrings();
    //Quick.sort(a);
    show(a);
    assert isSorted(a);

    // shuffle
    StdRandom.shuffle(a);

    for (int CUTOFF = 1; CUTOFF <= 25; CUTOFF++) {
        // Búa til slembiFylki
        double[] slembiFylki = new double[1000000];
        for (int i = 0; i < 1000000; i++) {
            slembiFylki[i] = StdRandom.uniform(n:1000000.0);
        }

        // Stopwatch stuff
        Stopwatch timer = new Stopwatch();
        Quick.sort(slembiFylki);
        double eTime = timer.elapsedTime();

        // Prentar tímann
        StdOut.println("CUTOFF = " + CUTOFF + ", Elapsed time = " + eTime);
    }
}
```

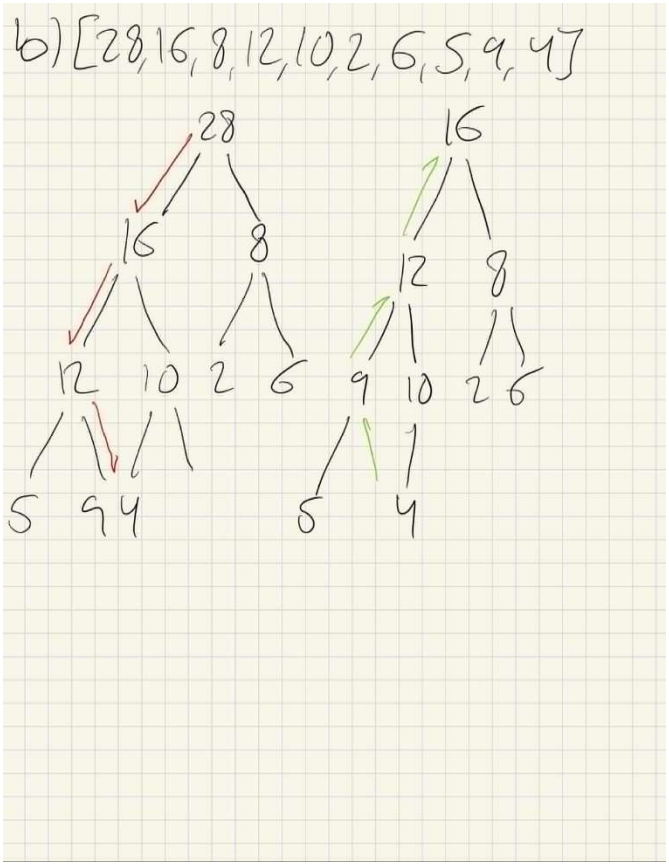
```
CUTOFF = 1, Elapsed time = 0.157
CUTOFF = 2, Elapsed time = 0.114
CUTOFF = 3, Elapsed time = 0.108
CUTOFF = 4, Elapsed time = 0.103
CUTOFF = 5, Elapsed time = 0.116
CUTOFF = 6, Elapsed time = 0.104
CUTOFF = 7, Elapsed time = 0.101
CUTOFF = 8, Elapsed time = 0.107
CUTOFF = 9, Elapsed time = 0.166
CUTOFF = 10, Elapsed time = 0.13
CUTOFF = 11, Elapsed time = 0.108
CUTOFF = 12, Elapsed time = 0.134
CUTOFF = 13, Elapsed time = 0.096
CUTOFF = 14, Elapsed time = 0.12
CUTOFF = 15, Elapsed time = 0.101
CUTOFF = 16, Elapsed time = 0.117
CUTOFF = 17, Elapsed time = 0.094
CUTOFF = 18, Elapsed time = 0.125
CUTOFF = 19, Elapsed time = 0.147
CUTOFF = 20, Elapsed time = 0.13
CUTOFF = 21, Elapsed time = 0.089
CUTOFF = 22, Elapsed time = 0.1
CUTOFF = 23, Elapsed time = 0.133
CUTOFF = 24, Elapsed time = 0.098
CUTOFF = 25, Elapsed time = 0.094
```

3. Fylkið er [2, 2, 5, 3, 6, 9, 7] eins og er. Spurningin er hvaða stök gætu hafa verið vendistakið fyrir Quicksort. Fyrir neðan verður svarað eftir röð, fyrsta stakið fyrst.
   1. 2 gæti verið vendistakið því öll stökin hægra megin við það eru jafnstór eða stærri.
   2. Sömu rök og fyrir stak 1. Vinstra megin við stakið er jafnstórt stak og hægra megin eru öll stök jafnstór eða stærri.
   3. 5 er ekki vendistak, það hefði víxlast við 3
   4. Sömu rök og fyrir 5, það hefði víxlast við 5
   5. 6 gæti verið vendistak því allt vinstra megin við 6 er minna og allt hægra megin er stærra
   6. 9 er ekki vendistak, það ætti að vera aftast
   7. 7 ætti að víxlast við 9

4. a) Myndin fyrir neðan sýnir hvert skref í innsetningunni. Lokafylkið er [17,11,15,2,9,8,4]



b)



5. Ég hef ákveðið að svara öllu í einni mynd.
   a) minnsta stakið þarf að vera neðst, því það getur ekki verið fyrir ofan neitt. Það verður að vera minnst og foreldri verður að vera stærra
   b) næst minnsta stakið getur ekki verið næst neðst nema í reit 6(eini reiturinn með bláu og grænu) vegna þess að bara eitt stak má vera minna og reitur 6 er sá eini sem er bara með eitt barn og þar með væri reitur 12 allra minnsta með næstminnsta sem foreldri.
   c) þriðja minnsta getur verið í hvaða reit sem er af 4-12. Allir þeir reitir bjóða uppá 2 eða færri börn sem er mikilvægt skilyrði fyrir max röðun. Ef þriðja minnsta er í 6 eða 7 hinsvegar, gætu næst minnsta og minnsta fundið sér hvað sem er af 8-12 og svo 6 ef þriðja minnsta er í 7 og vice versa.