Algoritmit ja tekoäly harjoitustyö

Viikkoraportti 2

Toinen viikko toi aikataulutushaasteita, mutta sain koodin rakentamisen aluilleen. Wikipediasta löysin Miller-Rabinin testiin pseudokoodin ja tein sillä ensimmäiset vedokset sekä todennäköisen alkuluvun testaamiseen, että todennäköisen n-bittisen alkuluvun luomiseen. Alku vaikutti lupaavalta, mutta kun ryhdyin

```
Input #1: n > 2, an odd integer to be tested for primality
Input #2: k, the number of rounds of testing to perform
Output: "composite" if n is found to be composite, "probably prime" otherwise

let s > 0 and d odd > 0 such that n − 1 = 2^s d  # by factoring out powers of 2 from n − 1
repeat k times:
    a ← random(2, n − 2)  # n is always a probable prime to base 1 and n − 1
    x ← a^d mod n
    repeat s times:
        y ← x^2 mod n
        if y = 1 and x ≠ 1 and x ≠ n − 1 then # nontrivial square root of 1 modulo n
            return "composite"
        x ← y
    if y ≠ 1 then
        return "composite"
return "probably prime"
```

Figure 1: Wikipedian pseudokoodi

tuottamaan isompia alkulukuja nuo a^d ja x^2 synnytti ylivuotoa jo 10-bittisillä alkuluvuilla. Lisäsin niihin enemmän modulaarisuutta ja nyt ainakin 24-bittinen onnistui yhdellä yrittämällä. Toki se rupesi ottamaan jo hetkisen. Pitänee tarkastella, onko tämä toteutettu järkevämmin jossain muualla. Automaattisten testien rakentaminen jäi nyt vielä tältä viikolta uupumaan. Loin alustavan käyttöliittymän, joka on tekstipohjainen ja toistaiseksi vaan omaan testailuun suunnattu. Alkulukutestiä ja etenkään faktorointia tuskin jätän käyttöliittymään myöhemmin.

Tapahtuma	Kuvaus	Aika (h)
koodauksen alotus	alkulukutesti ja käyttöliittymän ensimmäiset iteraatiot	1
koodausta	alkulukugeneraattori ja ongelmien löytyminen	0,5
koodausta	hyvin nukutun yön jälkeen ratkaisuyrityksen raapustelu	0,5
raportit ja git		1

Toki taulukko ei sisällä sitä asioiden passiivista sulattelua ja jäsentelyä