

Algoritmikerho 2.3.

Juha Harviainen

Alkuluvut

- **Alkuluvut:** Luvut jotka jaollisia vain 1:llä ja itsellään
- 2, 3, 5, 7, ...
- Olennainen osa monia lukuteoreettisia ongelmia

Tarkistus

```
def alkuluku(n):  
    if n <= 1:  
        return False  
    d = 2  
    while d < n:  
        if n % d == 0:  
            return False  
        d += 1  
    return True
```

Tarkistus

- Kaikkia lukuja n asti ei tarvitse tarkistaa
- Jos d jakaa n , niin $d * (n / d) = n$
- Joko d tai n / d on enintään $\sqrt{n}$
- **Riittää katsoa neliöjuureen asti!**

Tarkistus

```
def alkuluku(n):  
    if n <= 1:  
        return False  
    d = 2  
    while d * d <= n:  
        if n % d == 0:  
            return False  
        d += 1  
    return True
```

Eratostheneen seula

- Entä jos halutaankin tietää luvuille 1.. n ovatko alkulukuja?
- Eratostheneen seula

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Eratostheneen seula

- Entä jos halutaankin tietää luvuille 1.. n ovatko alkulukuja?
- Eratostheneen seula

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Eratostheneen seula

- Entä jos halutaankin tietää luvuille 1.. n ovatko alkulukuja?
- Eratostheneen seula

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Eratostheneen seula

- Entä jos halutaankin tietää luvuille 1.. n ovatko alkulukuja?
- Eratostheneen seula

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Eratostheneen seula

- Entä jos halutaankin tietää luvuille 1.. n ovatko alkulukuja?
- Eratostheneen seula

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Eratostheneen seula

- Entä ajankäyttö?
- Leikitään, että taulukko käydään läpi jokaiselle luvulle $1 \dots n$

$$\begin{aligned} & n/2 + n/3 + n/4 + n/5 + n/6 + n/7 + n/8 + \dots + n/n \\ & \leq n/2 + n/2 + n/4 + n/4 + n/4 + n/4 + n/8 + \dots + n/n \\ & = \quad n \quad + \quad \quad n \quad + \quad \dots \\ & \approx n \ln n \end{aligned}$$

Eratostheneen seula

- Samanlainen taktiikka toimii kaikkien tekijöiden kirjaamiseen

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Teoreettinen huomautus – lineaariseula

- Normaali seula käsittelee samoja lukuja useasti
 - 6:ssa käydään alkuluvuilla 2 ja 3

```
is_prime = [i > 1 for i in range(n+1)]
for i in range(2, n + 1):
    j = 2
    while i * j <= n:
        is_prime[i * j] = False
        j += 1
```

- Voidaanko käydä vain kerran?
- **Havainto:** Jokainen luku voidaan kirjoittaa yksikäsitteisesti muodossa $p * i$, missä p pienin luvun jakava alkuluku ja $p \leq i$

Teoreettinen huomautus – lineaariseula

- **Havainto:** Jokainen luku voidaan kirjoittaa yksikäsitteisesti muodossa $p * i$, missä p pienin luvun jakava alkuluku ja $p \leq i$

```
is_prime = [i > 1 for i in range(n+1)]
primes = []
for i in range(2, n + 1):
    if is_prime[i]:
        primes.append(i)
    for p in primes:
        if p * i > n: break
        is_prime[p * i] = False
        if i % p == 0: break
```

Teoreettinen huomautus – lineaariseula

- **primes** = [2]

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Teoreettinen huomautus – lineaariseula

- **primes** = [2,3]

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Teoreettinen huomautus – lineaariseula

- **primes** = [2,3]

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Teoreettinen huomautus – lineaariseula

- **primes** = [2,3,5]

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Teoreettinen huomautus – lineaariseula

- **primes** = [2,3,5]

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Teoreettisempi huomautus – alkulukujen määrä

- Alkulukujen lukumäärän voi laskea lineaarista nopeammin!
- Kerhokerran sivulla linkki Codeforces-blogiin aiheesta