

Stepenovanje

Ulaz

Data su dva prirodna broja n i k

Izlaz

n^k

Rešenje

Uobičajeno rešenje $O(k)$. Imamo li nešto efikasnije?

```
stepen(n, k)
{
    if (k == 1) {
        return n;
    } else {
        z = stepen(n, k / 2);

        if (k % 2 == 0) {
            return z * z;
        } else {
            return n * z * z;
        }
    }
}
```

Složenost

Broj množenja je $O(\log k)$

Zadaci

Zadatak 1

Kakva je veza između prethodnog algoritma i binarnog zapisa broja k ?

Rešenje:

Neka je $k = (k_{m-1}, k_{m-2}, \dots, k_0)_2$ binarni zapis broja k .

Rezultat inicijalno postavljamo na 1 ($z = 1$).

Važi sledeće:

$$k = (\dots(k_{m-1} \cdot 2 + k_{m-2} \dots) \cdot 2 + \dots) \cdot 2 + k_0$$

Prolazimo bitove broja k počev od indeksa $m-1$, preko $m-2, \dots$, do 0.

Tekući rezultat kvadriramo, a ako je $k_i = 1$ onda množimo i sa brojem n .

Dakle, $z := z^2 \cdot n^{k_i}$.

Zadatak 2

Prethodni algoritam ne minimizuje uvek broj množenja. Navesti primer izračunavanja n^k sa manjim brojem množenja.

Rešenje:

$$n^{15} = ((n^2 \cdot n)^2 \cdot n)^2 \cdot n - 6 \text{ množenja}$$

$$n^{15} = ((n^2)^2 \cdot n)^3 - 5 \text{ množenja}$$