Binarno iskanje za racionalna števila

Kratko sprotno poročilo o poteku dela

Ana Jurše

1 Uvod

Eden bolj osnovnih problemov v programiranju je zagotovo določanje neznanega celega števila x, $x \in \{1, ..., M\}$ z uporabo algoritma binarnega iskanja. Število potrebnih poizvedb v tem algoritmu je največ $log_2 M$.
Situacija pa je precej drugačna, če opazujemo iracionalna števila, saj števila x ne moremo določiti s končnim številom poizvedb.
Torej je bolj smisleno vprašanje koliko poizvedb potrebujemo za določitev pozitivnega racionalnega števila x, kjer sta števec in imenovalec celi števili omejeni z M.
V literaturi sem našla kar nekaj različnih idej in pristopov, moj izhodiščni članek pa opisuje algoritem, s časovno zahtevnostjo $\Theta(log_2 M)$, ki določi neznano število x v množici $Q_M = \{\frac{p}{q} : p, q \in \{1, ..., M\}\}$.

2 Algoritem

Vemo, da x lahko izrazimo kot $\lfloor x \rfloor + \frac{a}{b}$, kjer sta a in b tuji si števili in $a < b$.
Algoritem je sestavljen iz dveh delov.
V prvem delu določi $\lfloor x \rfloor$, z uporabo eksponentnega in binarnega iskanja, pri tem uporabi $log_2 \lfloor x \rfloor + O(1)$ poizvedb.
Težji del je določitev ulomka $\frac{a}{b}$. S pomočjo nekaj lem lahko pokažemo, da ga je možno določiti z največ $2log_2 M - 2log_2 \lfloor x \rfloor + O(1)$ poizvedbami.

3 Načrt dela

Projekta sem se lotila s pregledom različne literature in izhodiščnega članka. Za implementacijo algoritma predstavljenega v članku sem izbrala programski jezik Python.
Z robnimi primeri bom nato preverila časovno zahtevnost in pravilnost same implementacije. V nadaljevanju bi implementirala še katerega izmed algoritmov v literaturi in primerjala njihove časovne zahtevnosti v odvisnosti od M.