Binarni Euklidov algoritam

Domaća zadaća - Oblikovanje i analiza algoritama

Katarina Šupe

Prirodoslovno-matematički fakultet — Matematički odsjek

28. prosinca 2020.

Zadatak

Objasniti i implementirati efikasni Euklidov algoritam.
Program treba učitavati dva prirodna broja a i b za koje tražimo najveću zajedničku mjeru. Sam algoritam koristi njihov binarni zapis (jednostavno dijeljenje s dva - *shift* i sl.).

Najveća zajednička mjera

Definicija

Neka su a i b cijeli brojevi koji nisu oba nula. Za prirodan broj d kažemo da je najveća zajednička mjera (ili najveći zajednički djelitelj) brojeva a i b, i pišemo $d = NZM(a, b)$, ako d ima sljedeća svojstva:

1. $d \mid a$ i $d \mid b$
2. Za svaki prirodni c, $c \mid a$ i $c \mid b \Rightarrow c \mid d$

Kada su oba broja nula, svaki prirodni broj dijeli nulu, pa ne možemo primijeniti gornju definiciju. Stoga postavimo:

$$NZM(0, 0) = 0$$

Najveća zajednička mjera

Iz definicije slijedi:

$$NZM(a, b) = NZM(b, a)$$

$$NZM(a, b) = NZM(-a, b)$$

$$NZM(a, 0) = |a|$$

Još neka bitna svojstva:

$$NZM(k \cdot a, k \cdot b) = k \cdot NZM(a, b)$$

$$\text{ako je } NZM(a, b) = 1 \text{ tada je } NZM(a, k \cdot b) = NZM(a, k)$$

$$NZM(a, b) = NZM(a - b, b)$$

Najveća zajednička mjera

Iz osnovnog teorema aritmetike slijedi faktorizacija nekog prirodnog broja a (do na poredak prostih faktora):

$$a = 2^{a_2} \cdot 3^{a_3} \cdot 5^{a_5} \cdot 7^{a_7} \cdots = \prod_{p \text{ prime}} p^{a_p}$$

gdje su a_p jedinstveni nenegativni brojevi. Svi osim konačno mnogo eksponenata a_p su jednaki nula.
Po definiciji tada slijedi:

$$NZM(a, b) = \prod_{p \text{ prime}} p^{\min(a_p, b_p)}$$

Npr. $a = 24 = 2^3 \cdot 3^1$, $b = 4 = 2^2 \cdot 3^0$ pa je
$NZM(a, b) = 2^{\min(3,2)} \cdot 3^{\min(1,0)} = 2^2 \cdot 3^0 = 4$

Općenito o Euklidovom algoritmu

Gornji navedeni algoritam za pronalaženje najveće zajedničke mjere je jasan na papiru te vraća točan rezultat. Međutim, nije efikasan, jer zahtjeva faktorizaciju brojeva a i b, što zahtijeva dijeljenje tih brojeva s prostim brojevima redom, sve dok a i b ne postanu 1.
Euklid je u svojim *Elementima* pronašao metodu za određivanje najveće zajedničke mjere dvaju cijelih brojeva, bez faktoriziranja danih brojeva.

Općenito o Euklidovom algoritmu

Odsad možemo promatrati nalaženje najveće zajedničke mjere dvaju nenegativnih brojeva, što je u skladu s našim zadatkom. Spomenimo dvije verzije Euklidovog algoritma za nalaženje najveće zajedičke mjere, koji su doveli do efikasnog binarnog algoritma. Prva verzija nalazi $NZM(a, b)$ oduzimanjem, a druga dijeljenjem s ostatkom.

Euklidov algoritam s oduzimanjem

Euklidov algoritam s oduzimanjem rekurzivno traži *NZM* manjeg broja te razlike većeg i manjeg broja, sve dok ne postanu jednaki. Pogledajmo kako to izgleda u C-u.

```
unsigned int NZM(unsigned int a, unsigned int b)
{
    if( a == b )
    {
        return a;
    }
    else if(a > b)
    {
        NZM(a - b, b);
    }
    else
    {
        NZM(a, b - a);
    }
}
```

Euklidov algoritam s dijeljenjem s ostatkom

Dijelimo s ostatkom sve dok a ne postane djeljiv s b. Tada je posljednji ostatak različit od 0 upravo $NZM(a, b)$.

```
unsigned int NZM(unsigned int a, unsigned int b)
{
    if(a % b == 0)
    {
        return b;
    }
    else{
        return NZM(b, a % b);
    }
}
```

Binarni algoritam

Binarni GCD algoritam (*greatest common divisor* = najveći zajednički djelitelj = najveća zajednička mjera) poznat je kao *Stein*ov algoritam (*Josef Stein*, 1967.) ili binarni Euklidov algoritam. Taj algoritam nalazi najveću zajedničku mjeru dvaju nenegativnih brojeva. *Stein*ov algoritam koristi jednostavnije artimetičke operacije od običnog Euklidovog algoritma s dijeljenjem s ostatkom. Dijeljenje mijenja s aritmetičkim posmacima (*shift*), usporedbama i oduzimanjem.

Ideja binarnog algoritma

Pomoću svojstava najveće zajedničke mjere, dolazimo do algoritma:

1. Ako je a jednak b, tada je $NZM(a, b) = a$.
2. Ako je a jednak 0, tada je $NZM(a, b) = b$.
3. Ako je b jednak 0, tada je $NZM(a, b) = a$.
4. Ako su a i b parni, tada je $NZM(a, b) = 2NZM(a/2, b/2)$.
5. Ako je a paran i b neparan, tada je $NZM(a, b) = NZM(a/2, b)$.
6. Ako je a neparan i b paran, tada je $NZM(a, b) = NZM(a, b/2)$.
7. Ako su a i b neparni te $a > b$, tada je $a - b$ paran broj te je $NZM(a, b) = NZM((a - b)/2, b)$.
8. Ako su a i b neparni te $a < b$, tada je $b - a$ paran broj te je $NZM(a, b) = NZM((b - a)/2, a)$.

Primjer: $a = 49, b = 14$

Računamo $NZM(49, 14)$

1. Ako je a jednak b, tada je $NZM(a, b) = a$.
2. Ako je a jednak 0, tada je $NZM(a, b) = b$.
3. Ako je b jednak 0, tada je $NZM(a, b) = a$.
4. Ako su a i b parni, tada je $NZM(a, b) = 2NZM(a/2, b/2)$.
5. Ako je a paran i b neparan, tada je $NZM(a, b) = NZM(a/2, b)$.
6. Ako je a neparan i b paran, tada je $NZM(a, b) = NZM(a, b/2)$.
7. Ako su a i b neparni te $a > b$, tada je $a - b$ paran broj te je $NZM(a, b) = NZM((a - b)/2, b)$.
8. Ako su a i b neparni te $a < b$, tada je $b - a$ paran broj te je $NZM(a, b) = NZM((b - a)/2, a)$.

Primjer: $a = 49, b = 14$

Računamo $NZM(49, 7)$

1. Ako je a jednak b, tada je $NZM(a, b) = a$.
2. Ako je a jednak 0, tada je $NZM(a, b) = b$.
3. Ako je b jednak 0, tada je $NZM(a, b) = a$.
4. Ako su a i b parni, tada je $NZM(a, b) = 2NZM(a/2, b/2)$.
5. Ako je a paran i b neparan, tada je $NZM(a, b) = NZM(a/2, b)$.
6. Ako je a neparan i b paran, tada je $NZM(a, b) = NZM(a, b/2)$.
7. Ako su a i b neparni te $a > b$, tada je $a - b$ paran broj te je $NZM(a, b) = NZM((a - b)/2, b)$.
8. Ako su a i b neparni te $a < b$, tada je $b - a$ paran broj te je $NZM(a, b) = NZM((b - a)/2, a)$.

Primjer: $a = 49, b = 14$

Računamo $NZM((49-7)/2, 7) = NZM(21, 7)$

1. Ako je a jednak b, tada je $NZM(a, b) = a$.
2. Ako je a jednak 0, tada je $NZM(a, b) = b$.
3. Ako je b jednak 0, tada je $NZM(a, b) = a$.
4. Ako su a i b parni, tada je $NZM(a, b) = 2NZM(a/2, b/2)$.
5. Ako je a paran i b neparan, tada je $NZM(a, b) = NZM(a/2, b)$.
6. Ako je a neparan i b paran, tada je $NZM(a, b) = NZM(a, b/2)$.
7. Ako su a i b neparni te $a > b$, tada je $a - b$ paran broj te je $NZM(a, b) = NZM((a-b)/2, b)$.
8. Ako su a i b neparni te $a < b$, tada je $b - a$ paran broj te je $NZM(a, b) = NZM((b-a)/2, a)$.

Primjer: $a = 49, b = 14$

Računamo $NZM((21-7)/2, 7) = NZM(7,7) = 7$

1. Ako je a jednak b, tada je $NZM(a,b) = a$.
2. Ako je a jednak 0, tada je $NZM(a,b) = b$.
3. Ako je b jednak 0, tada je $NZM(a,b) = a$.
4. Ako su a i b parni, tada je $NZM(a,b) = 2NZM(a/2, b/2)$.
5. Ako je a paran i b neparan, tada je $NZM(a,b) = NZM(a/2, b)$.
6. Ako je a neparan i b paran, tada je $NZM(a,b) = NZM(a, b/2)$.
7. Ako su a i b neparni te $a > b$, tada je $a - b$ paran broj te je $NZM(a,b) = NZM((a-b)/2, b)$.
8. Ako su a i b neparni te $a < b$, tada je $b - a$ paran broj te je $NZM(a,b) = NZM((b-a)/2, a)$.

Implementacija binarnog algoritma

Ovaj algoritam naziva se **binarnim**, jer kao što smo već spomenuli, ne koristi obično dijeljenje brojeva, već samo dijeljenje s 2.

Dijeljenje s 2

Desni posmak (*right shift*, >> u `C-u`):

Neka je a nenegativni broj. Tada broj a dijelimo s 2 koristeći desnim posmakom za jedno mjesto, tj. a >> 1.

Provjera parnosti

Bitovna konjunkcija (`AND`, & u `C-u`):

Neka je a paran nenegativni broj. Tada je a & 1 jednako 0.

Neka je a neparan nenegativni broj. Tada je a & 1 jednako 1.

Implementacija binarnog algoritma

Pokažimo, za primjer korištenja bitovne konjunkcije i desnog posmaka, kako bi se u C-u, u rekurzivnoj verziji algoritma, implementirale 5., 6. i 7. grana algoritma:

```
else if( ((a & 1) == 0) && ((b & 1) != 0) )
{
        return binarni_nzm(a >> 1, b);
}
else if( ((a & 1) != 0) && ((b & 1) == 0) )
{
       return binarni_nzm(a, b >> 1);
}
else if( ((a & 1) != 0) && ((b & 1) != 0) && (a > b) )
{
       return binarni_nzm((a - b) >> 1, b);
}
```

Složenost

Algoritam zahtijeva $\mathcal{O}(n)$ koraka, gdje je n broj bitova većeg od dva broja, kako svaka dva koraka reduciraju barem jedan od operanada za barem faktor 2. Svaki korak uključuje samo par aritmetičkih operacija ($\mathcal{O}(1)$). Ukoliko su brojevi veličine riječi, svaka artimetička operacija je jedna strojna operacija pa je broj strojnih operacija reda $\log(\max(a, b))$. Asimptotska složenost ovog algoritma je $\mathcal{O}(n^2)$, kako aritmetičke operacije (oduzimanje i *shift*) uzimaju linearno vrijeme za proizvoljno velike brojeve.

Literatura

- URL: https://web.math.pmf.unizg.hr/~veky/em/vjezbe/nzm.html.
- URL: https://codility.com/media/train/10-Gcd.pdf.
- URL: https://www.geeksforgeeks.org/steins-algorithm-for-finding-gcd/.
- URL: https://en.wikipedia.org/wiki/Binary_GCD_algorithm.
- URL: https://www.cut-the-knot.org/blue/binary.shtml.
- URL: https://gmplib.org/manual/Binary-GCD.
- URL: https://radiusofcircle.blogspot.com/2016/10/binary-gcd-algorithm-implementation.html.
- Knuth, Donald E. *The Art of Computer Programming, Volume II: Seminumerical Algorithms, 2nd Edition*. Addison-Wesley, 1981. ISBN: 0-201-03822-6.