Algoritmi
Domača naloga 4

Sara Bizjak | 27202020

Maj 2021

Problem 1 – Najmanjši skupni množitelj in modularno potenciranje

A del: *Najmanjši skupni množitelj*

Definirajmo `lcm(`$a_1, a_2, \ldots, a_n$`)` kot najmanjši skupni množitelj n celih števil $a_1, a_2, \ldots, a_n$. Ta funkcija vrne najmanjše nenegativno celo število, ki je množitelj vseh števil a_i. Pokežmo, kako bi učinkovito izračunali `lcm(`$a_1, a_2, \ldots, a_n$`)` z uporabo funkcije `gcd`, ki je dvoparametrna.
Formula, ki povezuje funkciji za najmanjši skupni množitelj dveh števil in največji skupni delitelj dveh števil, je

$$\mathrm{lcm}(a, b) = \frac{ab}{\mathrm{gcd}(a, b)}$$

Ker zgornja zveza velja za dva parametra, funkcijo za izračun `lcm(`$a_1, a_2, \ldots, a_n$`)` definiramo rekuzrvino. Definirajmo funkcijo $\texttt{LCM}_{rek}$, ki za vhod velikosti 2 (torej za dve števili [a, b]), vrne kar $\mathrm{lcm}(a, b)$, če pa je vhodnih števil več, torej za vhodni seznam $[a_1, a_2, \ldots, a_n]$, pa se kliče rekurzivno na način

$$\texttt{LCM}_{rek}([\mathrm{lcm}(a_1, a_2), a_3, \ldots, a_n])$$

B del: *Modularno potenciranje*

Algoritem za modularno potenciranje je predstavljen v [[1], chapter 31]. Sestavimo algoritem za modularno potenciranje, ki b bitov obhodi iz desne proti levi namesto iz leve prosti desni.

Ideja je, da ko eksponent ni potenca števila 2 (preverimo tako, da pogledamo ostanek pri deljenju z 2), moramo vmesni rezultat dodatno pomnozimo z vrednostjo trenutne osnove (po modulu). Če pa je eksponent deljiv s številom 2, osnovo kvadriramo (po modulu) in eksponent razpolovimo.
Delujoča python koda je dostopna v priloženi datoteki `dn4.py`. Psevdo-koda opisanega algoritma:

```
def modularnoPotenciranje(a, b, n):
    """
    Vhod: a, b, n
    Izhod: a**b mod n
    """

    ce b == 0:
        vrni 1

    liha = 1                // zapisujemo liha mnozenja posebej
    osnova = a % n
    dokler b > 1:
        ce b % 2 == 1:                  // ce trenutni bit liho stevilo
            liha = (osnova * liha) % n
        sicer:                          // ce trenutni bit sodo stevilo
            b = b // 2
            osnova = (osnova ** 2) % n

    vrni (osnova * liha) % n
```

Problem 2 – problem trgovskega potnika

Podano imamo naslednjo hevristiko najbljižje točke za gradnjo poti trgovskega potnika, ki upošteva trikotniško neenakost.
Začnimo z izbiro poljubnega mesta v. Predpostavimo, da je to mesto legalen cikel (sam vase), ki predstavlja pot trgovskega potnika. Poiščimo mesto u, ki je najbljižje v in ga dodajmo v cikel tako, da u vstavimo v cikel takoj za v. Sedaj imamo cikel preko dveh mest. Ponovno poiščemo mesto u, ki je najbližje kateremukoli mestu na ciklu. Recimo, da je to najbližje mesto iz cikla mesto v. Ponovno dodajmo u takoj za v in to ponavljajmo dokler še imamo mesta izven cikla. Pokažimo, da je ta hevristika dvo aproksimacijska.

Literatura

[1] Thomas H. Cormen. *Introduction to algorithms*. MIT press, 2009.