Algoritmi
Študijsko leto 2021/2022

4. domača naloga

Poročilo domače naloge

Ivo Pajer
Vpisna št. 63180218

Ljubljana, 13. maj 2022

Kazalo

1 Problem 1 - Najmanjši skupni množitelj in modularno potencirajnje

1.1 A - Najmanjši skupni množitelj

Definirati moramo $lcm(a_1, a_2, ..., a_n)$, ki nam pove najmanjši skupni množitelj števi $a_1, a_2, ..., a_n$, z uporabo že definirane funkcije $gcd(a, b)$, ki nam pove največji skupni deljitelj a in b.
Najmanjši skupni množitelj nam pove najmanjše število, ki ga nek nabor deli brez ostanka. Pri dveh številih, ki sta si tuji, je njun $lcm()$, njun produkt. Če pa si števili nista tuji, pa je $lcd(a, b) = \frac{a*b}{gcd(a,b)}$. Za več števil, lahko funkcijo definiramo rekurzivno kot $lcm(...(lcm(lcm(a_1, a_2), a_3), ...), a_n)$. Na vsakem koraku izvedemo poizvedbo po $lcm()$ z uporabo funkcije $gcd()$ in množenja izbranih števil.

1.2 B - Modularno potenciranje

Naša naslednja naloga je, da algoritem predstavljen v knjigi [1]. Program moramo pretvoriti tako, da deluje iz desne proti levi namesto leve proti desni.
Psevdokoda algoritma se glasi:

```
function ModExp(a, b, n)
    if n == 1 then
        return 0
    else
        rez = 1
        a = a mod n
        while b > 0 do
            if b mod 2 == 1 then
                rez = (rez*a) mod n
            end if
            b = b > > 1
            a = (a*a) mod n
        end while
        return rez
```

2 Problem 2 - Problem trgovskega potnika

2. Problem nam govori o problemu potujočega potnika (Travelling Salesman problem), z podano hevristiko moramo pokazati, da je ta hevristika dvo aproksimacijska, kar pomeni, da je rešitev algoritma kvečemu 2-krat slabša od optimalne rešitve.
Problem lahko prevedemo na iskanje Hamiltonskega cikla z minimalno ceno, kot je pokazano v poglavju 35 [1]. Trikotniška neenakost nam pove, da mora za vsako vozlišče v grafu **G**, veljati: $c(u, w) \leq c(u, v) + c(v, w)$, kjer $c(u, w)$ - cena poti iz u v w, ter $u, v, w \in \mathbf{G}$.
Zgornjo metodologijo uporabimo, da konstruiramo minimalno vpeto drevo, čigar cena nam poda spodnjo mejo za optimalno rešitev problema. Nato to drevo uporabimo za konstrukcijo sprehoda, ki je maksimalno dvakrat daljši, kot cena vpetega drevesa, kar nam omogoča trikotniška neenakost.
V algoritmu bomo uporabili še funkcijo $MST - PRIM(G, c, r)$, ki nam konstruira minimalno vpeto drevo po Primovem algoritmu. Psevdokoda se glasi:

```
function TOUR(G, c)
    Izberi vozlišče r ∈ G, naj bo korensko vozlišče
    izračunaj minimalno vpeto drevo z uporabo MST − PRIM(G, c, r)
    Naj bo H seznam vozlišč, ki so razporejena po zaporedju obiska
    return Hamiltonski cikel H
```

2.1 Dokaz o dvo-aproksimaciji

Z enostavno implementacijo $MST - PRIM$, lahko dosežemo časovno zahtevnost $\Theta(V^2)$. Sedaj pa bomo pokazali še, da algoritem vrne sprehod, ki je največ 2-krat daljša od optimalne.
Naj H^* označuje optimalno pot čez **G**. Kot smo že omenili, nam vpeto drevo da spodnjo mejo za optimalno pot. Polna pot vpetega drevesa T nam našteje vozlišča, ko jih prvič obiščemo in ko se v njih vrnemo, da se pomaknemo v pod-drevo. Polno pot bomo imenovali W. Ker polna pot vsako vozlišče prepotuje največ 2-krat, imamo $c(W) = 2c(T)$. Neenakost in enačbi pa nam skupaj povesta, da $c(W) \leq 2c(H^*)$, kar pomeni, da je cena W največ 2-kratnik optimalne poti.
Polna pot pa vozlišča lahko obišče večkrat, kar ga optimalna pot (ali pa sprehod ne more). Lahko pa zaradi trikotniške neenakosti kadarkoli izbrišemo katerokoli vozlišče iz W in se cena ne zviša. Če postopoma izvajamo to operacijo, lahko izbrišemo iz W vse naknadne obiske vozlišč. Če je H cikel, ki pripada temu sprehodu, in je H zgrajen iz vozlišč iz W, lahko sklepamo,

da je H manjši ali enak W, zato lahko privzamemo, da $c(H) \leq c(W)$.

Literatura

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition.* The MIT Press, 3rd ed., 2009.