The triangle scheduling problem

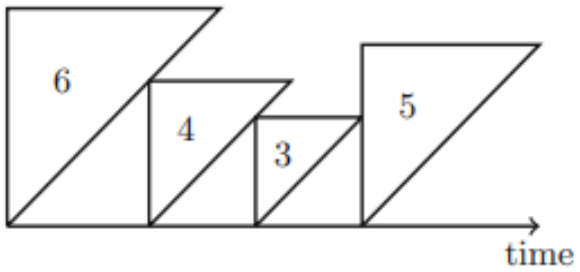
Arh Blaž, Matušek Matic

9. 11. 2022

1 Uvod

1.1 Problem

Imamo n pravokotnih enakokrakih trikotnikov, ki so določeni z dolžino kraka d_i za $i = 1, 2, ..., n$ in pravim kotom med krakoma. Trikotnike postavimo, z nogo na x-os, na sledeči način.



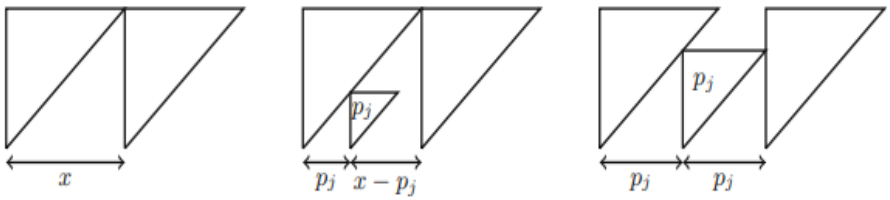
Trikotniki morajo biti postavljeni tako, da je hipotenuza trikotnika naraščajoča glede na x-os.
Želimo si trikotnike postaviti tako, da je dolžina teh trikotnikov najkrajša. Trikotniki se med seboj ne smejo prekrivati, prav tako pa mora biti krak noge pravokoten na x-os.

1.2 Uporabnost

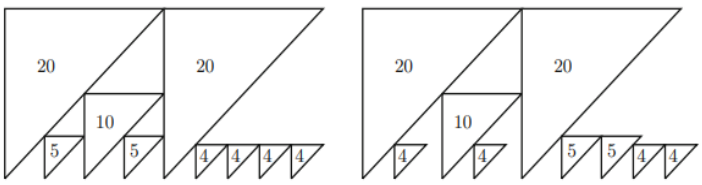
Želimo si sestaviti urnik iz n opravil. Naj bo d_i pomembnost i-tega opravila. Naš cilj je sestaviti čim krajši urnik. Želimo, da se pomembnejša opravila izvedejo, v primeru zakasnitve le teh pa lahko manj pomembna opravila izpustimo. Večji kot je trikotnik, bolj pomembno je opravilo. Kot je razvidno iz zgornje slike, lahko opravila z manjšo pomembnostjo vstavimo pod opravila z večjo pomembnostjo. To pomeni, da manjše trikotnike vstavljamo pod večje. Torej, če pride do zakasnitve pomembnejšega opravila (večjega trikotnika), manj pomembno opravilo izpustimo (manjši trikotnik).

2 Greedy algoritem

Greedy algoritem je kateri koli algoritem, ki rešuje problem tako, da na vsakem koraku naredi lokalno optimalno izbiro. Greedy algoritem najprej razvrsti trikotnike tako, da velja
$d_1 \geq d_2 \geq d_3 \geq ... \geq d_n$. Prvi trikotnik razvrsti na $x = 0$, kar ustvari prostor pod trikotnikom 1. Nato vsak trikotnik $j = 2, ..., n$ postavi v največji možen prostor pod že postavljenimi trikotniki. Če je več enakih prostorov, izbere prvega povrsti. Če ima izbrani prostor pod trikotnikom dolžino l in se začne v času x_i, potem je trikotnik j postavljen na mesto $x_j = x_i + d_j$. Če je $2d_j \geq l$, potem se vsi trikotniki k, za katere je $x_k \geq x_j$ zamaknejo za $2d_j - l$, da se ne prekrivajo. Opisano dogajanje prikazuje spodnji primer($x = l, p_i = d_i$).



Pri številnih problemih Greedy algoritem ne poišče optimalne rešitve, nič drugače ni pri našem problemu. Na naslednji sliki prikažemo primer, ko Greedy ni optimalen. Greedy algoritem (desna slika) vrne dolzino 42 in ne vrne optimalne dolžine, ki je 40 (leva slika).



Hitrost Greedy algoritma je polinomska. Da se pokazati, da je aproksimacijsko razmerje zgornje meje točnosti Greedy algoritma 1.5 optimalnega časa. Torej, če je optimalni čas določenega primerna 40 časovnih enot, potem Greedy pokaže največ 60 časovnih enot.

3 Načrt dela

Za zdaj sva naredila program, ki ustvari trikotnike in jih razporedi na x-os. Prav tako sva naredila brute force algoritem, da lahko svoje rezultate preveriva.
V nadaljevanju bova sprogramirala še:

- Greedy algoritem in
- algoritem, ki poišče optimalno rešitev hitreje kot brute force algoritem.