

# Število podpoti pri grafih z danim ciklomatskim številom

*Poročilo projektne naloge  
Skupina 18: Maja Košmrlj, Jan Maradin*

## vprašanja za profesorja

- ali lahko midva brez geodetskih podpoti, ker imava občutek da nama bo časa zmanjšalo še posebej zaradi tega da vse tako dolgo traja
- kakšno temperaturo morava uporabljati, sedaj imava delala predvsem
- ali je okej, da midva napiševa kodo, tako da ko jo ponovno poženeva posodobi podatke, torej da večkrat ko poženeva kodo boljši rezultat najde,  
ali je poanta, da najdeva midva način, kako z dobrimi približki prideva že od enem pogonu do dobrega približka
- kakšno analizo morava narediti, ali lahko damo v R v tabelo ali v excel?

## Najini podvigi

- 
- ko sva probala za  $n = 9$  in  $N = 4000$  izvesti po Janovi metodi smo porabili do mu= 18 je rabil 20 minut kar je očitno čisto preveč počasi, saj bi s tem tempom delal za vse mu pri  $n = 9$  rabil po vsej verjetnosti več ur
- Zagnala sva iskanje minimuma za  $n = 9$  po Janovi metodi z nastavitevami `max_steps` = 300, `T_start` = 3, `T_end` = 0.001, `cooling` = 0.99. Računanje je trajalo približno 32 minut.  
naslednji ko sva pognala je trajalo 41 min (opomba: runnala sta dve funkcije hkrati)  
Problem: ker za  $n = 9$  ne začnemo več z dobrim približkom iz  $n = 8$ , se zgodi, da se pri dodajanju ene povezave ostale povezave ne spremenijo in se vrednost `subpath_number` sploh ne izboljša. Sklepava, da je težava v premajhnem številu korakov, da bi metoda našla boljši rezultat, torej da sva obstala v nekem lokalnem minimumu.
- Sedaj bova za  $n = 9$  uporabila še metodo za manjše grafe: za vsak par  $(n, \mu)$  poiščeva vse možne povezane grafe, za vsakega izračunava najmanjši `subpath_number` in nato shraniva tistega z najmanjšo vrednostjo (če bodo rezultati še izvedljivi v razumnem času).  
v 80 minutah je funkcija, ki generira vse grafe dokončala izračune za vse grafe do vključno  $n = 9$   
ogotovitev: srednje vrednosti se precej razlikujejo, ampakk ne ostalo je pa obiskuje

- sedaj poganjava kodo še za max, tako da vpisuje v isti csv kot sva že imela rezultate za min to delava v PRAVILNO2, trajalo je 37 min
- Pognala sva kodo `vecji_grafi` za  $n = 8$ , da bova rezultate primerjala z različico `_live`. Izvajala sva jo za minimum in maksimum ter za različno število ponovitev (300 in 4000). Prav tako sva v kodo dodala možnost nastavljanja  $N$  za poljuben  $n$ , tako da program sam izračuna  $\mu_{\max, \text{prev}}$  in  $\mu_{\max, n}$ .
   
za max je trajalo 8 minut pri 4000
   
za max T=30 N= 300 — 3,5min
   
za min T= 6 N= 2000 —
- Sedaj bova poskusila kodo pospešiti tako, da jo zaženeva za več vrednosti  $\mu$  hkrati (da bolje izkoristiva 100% CPU-ja) ter obenem povečava število korakov, tako da bo čas izvajanja ostal približno enak.

## Problemi

- ali koda ki ne zapisuje pravilno dela bolje od tista v pravilno? ali je bilo takrat samo naključje

## to do— če bo čas

- poženi za  $n = 9$  še za 2000
- prveri dodatne lastnosti v grafih

## game plan

- Za  $n = 9$  uporabiva rezultate iz `ne_live_8`.
- Napiši funkcijo, ki izračuna minimum in maksimum hkrati za poljuben  $n$ .
- Izboljšaj izkoriščenost CPU-ja v tej funkciji.
- Zapiši funkcijo, ki bo rezultate naključnih grafov za  $n = 8, 9, 10$  zapisovala v CSV.
- Opraviva analizo podatkov.
- Zapiši funkcijo, ki bo rezultate kaktusnih grafov za  $n = 8, 9, 10$  zapisovala v CSV.
- preverit je treba če so slučajno ptc grafi pri majhnih grafih