SKUPINA:

* Urša Kumelj in Miha Rakovec
* 3. naloga pri Dinamičnem programiranju na strani 124 in 1. naloga o Najkrajših poteh na strani 297

OPIS TEME:

* Pri 3. nalogi, ki v a) točki zahteva, da se poišče največjo vsoto podseznama, bi uporabila tehnike dinamičnega programiranja, pod katerimi je tudi naloga zapisana. Kot že zapisano v knjigi je dinamično programiranje rekurzija brez ponavljanja. Večina se nas, pri takih vrstah nalog, odloči za delo s tabelami, saj je bolj enostavno. Bolj pravilno pa je reševanje z rekurzijo, kar je seveda bolj zapleteno. Algoritmi dinamičnega programiranja se rešujejo na dva načina:
  + Formulizacija problema rekurzivno. Zapišemo rekurzivno formulo ali algoritem glede na odgovore na manjše podprobleme.
  + Zapis rešitve rekurzije od spodaj navzgor. Torej, da zapišemo algoritem, ki se začne z osnovnimi primeri naše rekurzije in se prebija do končne rešitve z upoštevanjem vmesnih podproblemov v pravilnem vrstnem redu.
* 1. naloga iz poglavja o Najkrajših poteh od nas zahteva, da izračunamo, koliko časa bi potrebovali za izračun najkrajše poti med dvema ogliščema u in v, v tako imenovanem 'looped tree' z n vozlišči, z uporabo Dijkstra-novega algoritma. Dijkstrov algoritem je algoritem za iskanje najkrajših poti med vozlišči v uteženem grafu. Primer uteženega grafa bi si lahko predstavljali, kot da vodimo letalsko družbo in bi radi model, ki nam bo pomagal oceniti stroške goriva za poti, kjer letimo. V tem primeru bi bila vozlišča letališča, robovi bi predstavljali lete, utež roba pa bi bila cena goriva, porabljena za let. Dijkstrov algoritem bi nam pri tem modelu vrnil najcenejšo pot.

VIRI:

* <https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm#Running_time>
* <https://en.wikipedia.org/wiki/Dynamic_programming>
* [Jeff Erickson, Algorithms, June 13, 2019](https://ucilnica.fmf.uni-lj.si/pluginfile.php/127554/mod_resource/content/0/Algorithms-JeffE_jun2019-v2.pdf)