

PROJEKTOVANJE I ANALIZA ALGORITAMA

Laboratorijska vežba 3
Ristovski Nikola 19347
Tehnologija : C#

Greedy vs Dynamic rešenje

Ovu laboratorijsku vežbu odradio sam na dva načina, tj primenom dva tipa algoritama, dinamički i pohlepni (greedy). Greedy algoritmi gledaju da do rešenja određenog problema dođu efikasno, procenjujući u svakom koraku najbolji potez tj odluku, ne razmatrajući prethodne odluke. Zato oni ne daju uvek apsolutno tačno rešenje, ali je ono zadovoljavajuće. To uočavam pri testiranju mog algoritma, a kao referentnu tačku odlučio sam da uzmem upravo algoritam odrađen dinamičkim tipom, koji daje tačnija rešenja (kažem tačnija, a ne tačna, jer neće ni ovaj algoritam dati uvek tačno rečenje, o tome ću kasnije, ali u većini slučajeva daje tačno rešenje pa je dobar kao tačka poređenja).

Implementacija rešenja

Greedy algoritam

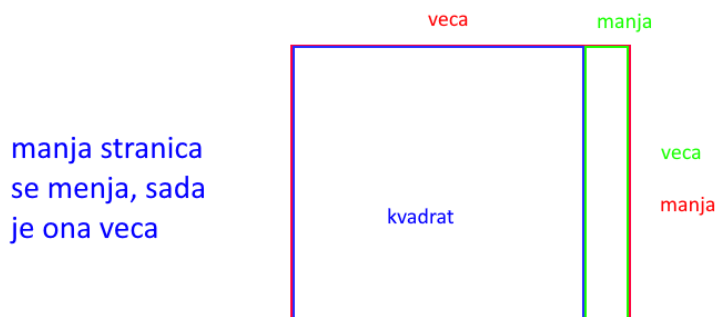
Ovaj algoritam se bazira na sledećem principu: pravougaonik ima veću i manju stranicu pa možemo odseći kvadrat dimenzije manje stranice. Nakon toga nam ostaje opet pravougaonik (ili kvadrat), pa nad njim ponovimo postupak. Imamo dva slučaja koja su od značaja jedino u kodu, kada je neophodno promeniti koja je veća a koja manja stranica, a to se može videti na sledećoj slici.

U prvom slučaju na slici vidimo da manja stranica i nakon odsecanja kvadrata ostaje manja relativno na drugu (onu koja je dobijena kao `vecaStranica` – `manjaStranica` starog pravougaonika).

U drugom slučaju, tj na slici ispod, vidimo da nakon odsecanja kvadrata sada manja stranica postaje veća u odnosu na novodobijenu (`vecaStranica` – `manjaStranica` starog pravougaonika).

Ove informacije su, kao što rekoh, jedino bitne da bismo u kodu razumeli deo:

```
(vecaStranica, manjaStranica) = (manjaStranica, vecaStranica);
```



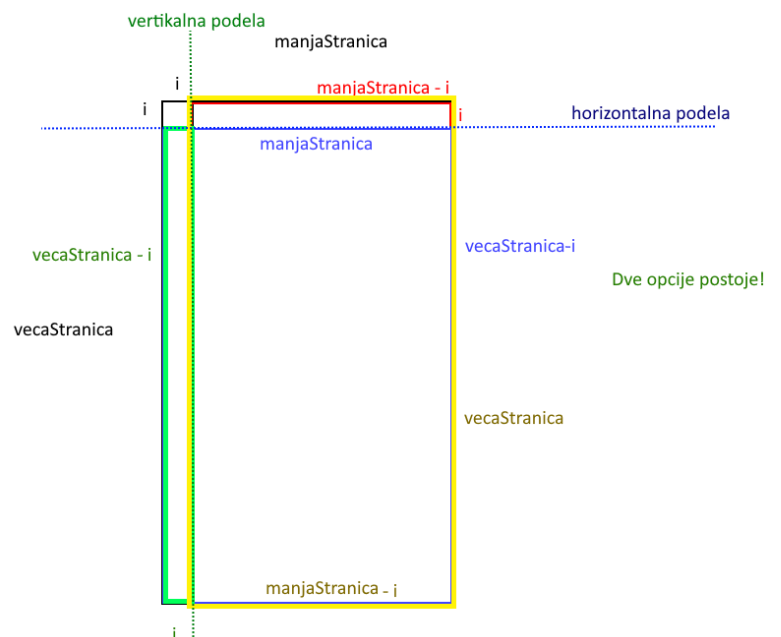
Dynamic algoritam

Ovo rešenje se bazira na sledećem razmišljanju:

Iz pravougaonika možemo odseći kvadrat dimenzije $i \times i$, gde i može biti 1, 2, 3... do k gde je k dimenzija manje stranice pravougaonika. Sa slike možemo da vidimo da imamo dve situacije:

1. Kada horizontalno podelimo ostatak figure (bez isečenog kvadrata) na dva nova pravougaonika (plava isprekidana linija)
2. Kada vertikalno podelimo isti taj ostatak figure na dva nova pravougaonika (zeleno isprekidana linija)

Prvo ćemo rekursivno pozvati nalaženje minimalnog broja kvadrata koje možemo iseći iz oba nova pravougaonika ako smo uzeli situaciju da je horizontalno podeljen ostatak (plavi i crveni pravougaonici na slici). Zatim uradimo isto to, rekursivno nalaženje broja kvadrata iz novih kvadrata ako je ostatak figure isečen vertikalno (dakle u žutom i zelenom pravougaoniku na slici). Zatim poredimo rešenja koja smo dobili i odaberemo onu podelu (horizontalnu ili vertikalnu) kojom bismo dobili najmanje kvadrata.



Kako se porede rešenja

U nekim situacijama se greedy algoritam poklapa sa dinamičkim, međutim u većini daje gora rešenja (da je potrebno više kvadrata nego što zaista jeste). Međutim ako pogledamo samu implementaciju algoritama shvatamo da je mnogo razumnija i efikasnija greedy varijanta, za razliku od dinamičke koja je rekursivna pa je pozivamo mnogo puta, a i neophodno je imati dodatnu strukturu koja će čuvati rešenja potproblema, da ih ne bismo rešavali mnogo puta, već samo jednom (kod mene u kodu koristio sam rečnik).

U ovoj situaciji pravougaonik je 6×7 . Dinamičko rešenje je očigledno tačno, dok je greedy približno ali ne i tačno.

```
=====
-> Unesite dimenzije pravougaonika:
      a -> 6
      b -> 7
-> Testiramo [DYNAMIC] verziju algoritma:
      -> Pravougaonik: 6 x 7 ----> 5 kvadrata.
      -> [4x4] [2x2] [2x2] [3x3] [3x3]
-> Testiramo [GREEDY] verziju algoritma:
      -> Pravougaonik: 6 x 7 ----> 7 kvadrata.
      -> [6x6] [1x1] [1x1] [1x1] [1x1] [1x1] [1x1]
=====
```

U ovom slučaju ipak rešenja se poklapaju, tako da je moguće da i greedy da tačno rešenje.

Ako uzmemo pravougaonik većih dimenzija jasno nam je koliko je odstupanje greedy i dinamičkog rešenja ogromno.

Medjutim u nekim situacijama čak i dinamičko rešenje nije potpuno tačno. Uzmimo kao primer pravougaonik dimenzija 13x11. Sa leve strane vidimo da je rešenje koje dobijamo 8. Medjutim, u stvarnosti je rešenje 6. Ovo je očigledno ako uporedimo način na koji dinamički algoritam radi, i kako su na slici ispod isečeni kvadrati.