

Amortizirana analiza

Tomaž Dobravec, Algoritmi in podatkovne strukture 2

Amortizirana analiza

- ✧ Amortizirana analiza – pristop k ocenjevanju zahtevnosti algoritmov in podatkovnih struktur.
- ✧ Pogosto se zgodi, da je “worst-case” analiza (po krivici) preveč pesimistična.
- ✧ Amortizirano analizo uporabljamo za ocenjevanje zahtevnosti postopkov (bodisi algoritma bodisi operacij v podatkovnih strukturah)
- ✧ Amortizirana analiza ne ocenjuje **posameznih operacij** postopka, rezultat analize je ocena **celote!**
 - čeprav so nekatere operacije postopka morda počasne, obstajajo tudi druge, ki so hitre;
 - počasne in hitre operacije postopka skupaj v povprečju morda niti niso tako slabe.
- ✧ Amortizirana analiza skuša ceno dragih operacij postopka enakomerno porazdeliti med poceni operacije.
- ✧ Amortizirana analize ocenjuje zaporedje **n** operacij nekega postopka.
- ✧ Recimo, da imamo končen nabor operacij $O = \{o_1, o_2, \dots, o_k\}$. Zahtevnost operacije $o \in O$ označimo s T_o .
 - Worst case analiza
 - Verjetnostna analiza
 - Amortizirana analiza

Vrste amortizirane analize in primeri

✧ Amortizirano analizo lahko izvajamo na več načinov, najpogosteje z uporabo ene izmed naslednjih metod:

- Metoda vsote (aggregate method)
- Metoda kopičenja - bankirski pristop (accounting method)
- Metoda potenciala - fizikalni pristop (potential method)

✧ Prikaz delovanja posamezne metode bomo izvedli na dveh primerih:

- Dvojiški števec

k-bitno polje $A[0, 1, \dots, k-1]$, vrednost števca:

$$x = \sum_{i=0}^{k-1} A[i] \cdot 2^i$$

operacija `povecaj()` števec poveča za 1

- Razširjen sklad

`push(S, x)`

`pop(S)`

`pop(S, k)`

Amortizirana analiza - metoda vsote

- ✧ Clij : na konkretnem primeru in za konkretno funkcijo $T(n)$ pokažemo, da je za vsak n cena zaporedja n operacij v najslabšem primeru enaka $T(n)$.
- ✧ V najslabšem primeru je potem povprečna cena (amortizirana cena) posamezne operacije $T(n)/n$.
- ✧ Pri tej metodi se amortizirana cena nanaša na **VSE operacije** (vse imajo enako povprečno ceno); ne razlikujemo med različnimi operacijami (ena je dražja, druga cenejša).
- ✧ **Povzetek: z metodo vsote seštejemo prispevke VSEH operacij in jih delimo z n .**

Metoda vsote – primer (razširjen sklad)

A series of horizontal blue lines for writing, with a vertical red margin line on the left side.

Metoda vsote – primer (dvojiški števec)

A grid of horizontal blue lines for writing, with a vertical red line on the left side. The grid consists of 20 horizontal lines and one vertical line on the left, creating a series of columns for writing.

Amortizirana analiza - metoda kopičenja

✧ Metoda kopičenja "obiskuje" operacije po vrsti od prve proti zadnji (n-ti).

✧ Vsaki obiskani operaciji metoda "zaračuna" njen obstoj v zaporedju.

✧ Cena, ki jo mora operacija plačati, se imenuje "amortizirana cena".

✧ Amortizirana cena operacije je lahko **višja** ali **nižja** od realne cene operacije.

razlika se shrani (**kredit**)

razlika se krije iz kredita prejšnjih operacij

✧ Amortizirane cene posameznih operacij pri tej metodi se med seboj lahko razlikujejo.

✧ Pomembno: za vsa zaporedja dolžine n mora veljati
$$\sum_{i=1}^n c_i \leq \sum_{i=1}^n \hat{c}_i$$

c_i označuje realno ceno i -te operacije, \hat{c}_i pa njeno amortizirano ceno

Metoda kopičenja – primer (razširjen sklad)

A series of horizontal blue lines for writing, with a vertical red margin line on the left side.

Metoda kopičenja – primer (dvojiški števec)

A series of horizontal blue lines for writing, with a vertical red line on the left side.

- ✧ Cilj: vsota amortiziranih cen zaporedja operacij je zgornja meja za vsoto realnih cen.
- ✧ Ne beležimo kredita posamezne operacije ampak le skupen kredit (potencial) celotne podatkovne strukture.
- ✧ Oznake: začetno stanje podatkovne strukture: D_o , stanje po i-ti operaciji: D_i . Realna cena i-te operacije c_i .
- ✧ Funkcija Φ – potencial (realno število) podatkovne strukture.
 $\Phi(D_i)$... potencial podatkovne strukture po i-ti operaciji.
- ✧ Amortizirana cena i-te operacije \hat{c}_i je definirana z $\hat{c}_i =$
- ✧ Amortizirana cena za zaporedje n operacij je zato:

$$\sum_{i=1}^n \hat{c}_i =$$

- ✧ Če je funkcija potenciala definirana tako, da je $\Phi(D_n) \geq \Phi(D_o)$, potem je **amortizirana cena zaporedja** zgornja meja za worst-case realne cene zaporedja.

- ✧ Ker v praksi ne vemo, koliko operacij bo v zaporedju, je praktična zahteva še $\Phi(D_i) \geq \Phi(D_o)$ za vsak i ; v tem primeru namreč vemo, da smo vedno plačali dovolj za naslednjo operacijo.
- ✧ Čeprav je amortizirana cena odvisna od izbire funkcije Φ , je ob upoštevanju zgornjih zahtev vedno ZGORNJA meja za realno ceno zaporedja; ob smiselni in preišljeno izbrani funkciji potenciala, se zgornja meja lahko približa tesni meji.

Povzetek: za poljubno realno funkcijo Φ , ki določa "potencial" podatkovne strukture, in za katero velja $\Phi(D_i) \geq \Phi(D_o)$ za vsak i , lahko izračunamo amortizirane cene za posamezne operacije. Če smo imeli pri izbiri funkcije Φ "srečo", bodo tako naračunane amortizirane cene in njihova najslabša vsota prinesli pametno oceno za zgornjo mejo.

Metoda potenciala– primer (razširjen sklad)

A series of horizontal blue lines for writing, with a vertical red line on the left side.

Metoda potenciala– primer (dvojiški števec)

A series of horizontal blue lines for writing, starting from the title and extending down the page.