Algoritmusok és adatszerkezetek II.

Amotrizált költségelemzés, Fibonacci kupacok

Szegedi Tudományegyetem

Amortizált költségelemzés

- A legrosszabb költségelemzés túl pesszimista tud lenni
- Amortizált költségelemzésnél az adatszerkezetek ‘életútját’ vizsgáljuk
- Lehetnek költséges műveleteink, ha azok *kellően* ritkák
  - Pl. dinamikusan bővülő tömb

Amortizált költségelemzés

- A legrosszabb költségelemzés túl pesszimista tud lenni
- Amortizált költségelemzésnél az adatszerkezetek ‘életútját’ vizsgáljuk
- Lehetnek költséges műveleteink, ha azok *kellően* ritkák
  - Pl. dinamikusan bővülő tömb

Fontos

Ennél az elemzésnél a véletlennek nincs szerepe: az egyes műveletek átlagos költségére adunk felső korlátot a *legrosszabb esetben*.

Amortizált költségelemzés

- A legrosszabb költségelemzés túl pesszimista tud lenni
- Amortizált költségelemzésnél az adatszerkezetek 'életútját' vizsgáljuk
- Lehetnek költséges műveleteink, ha azok *kellően* ritkák
    - Pl. dinamikusan bővülő tömb

Fontos

Ennél az elemzésnél a véletlennek nincs szerepe: az egyes műveletek átlagos költségére adunk felső korlátot a *legrosszabb esetben*.

Fő megközelítések

1. Összesítéses elemzés
2. Könyvelési módszer
3. Potenciálmódszer

Bináris számláló növelése

```
NÖVEL(A) {
  i=0
  while i < A.hossz és A[i] = 1 {
     A[i] = 0
     i = i+1
  }
  if i < A.hossz {
     A[i] = 1
  }
}
```

Bináris számláló növelése

```
NÖVEL(A) {
  i=0
  while i < A.hossz és A[i] = 1 {
     A[i] = 0
     i = i+1
  }
  if i < A.hossz {
     A[i] = 1
  }
}
```

i	3	2	1	0	$\sum$ktg.
	0	0	0	0	0
	0	0	0	1	1
	0	0	1	0	3
	0	0	1	1	4
	0	1	0	0	7
	0	1	0	1	8
	0	1	1	0	10
	0	1	1	1	11
	1	0	0	0	15
	1	0	0	1	16
	1	0	1	0	18
	1	0	1	1	19
	1	1	0	0	22
	1	1	0	1	23
	1	1	1	0	25
	1	1	1	1	26

Összesítéses elemzés

n hosszú műveletsorra állítunk föl $T(n)$ felső korlátot
$\rightarrow$ a műveletek átlagos költsége $T(n)/n$

Példa

k bites számlálón Növel művelet n-szeri végrehajtása: $O(nk)$
Helyes, de nem éles korlát, mivel az i pozíciójú bit csak minden 2^i számú végrehajtás után változik.

Amortizált költségelemzés – összesítéses elemzés

Összesítéses elemzés

n hosszú műveletsorra állítunk föl $T(n)$ felső korlátot
$\rightarrow$ a műveletek átlagos költsége $T(n)/n$

Példa

k bites számlálón Növel művelet n-szeri végrehajtása: $O(nk)$
Helyes, de nem éles korlát, mivel az i pozíciójú bit csak minden 2^i számú végrehajtás után változik.

Összesítéses elemzés

n hosszú műveletsorra állítunk föl $T(n)$ felső korlátot
$\rightarrow$ a műveletek átlagos költsége $T(n)/n$

Példa

k bites számlálón Növel művelet n-szeri végrehajtása: $O(nk)$
Helyes, de nem éles korlát, mivel az i pozíciójú bit csak minden 2^i számú végrehajtás után változik.

Élesebb korlát

$$\sum_{i=0}^{k-1} \left\lfloor \frac{n}{2^i} \right\rfloor < n \sum_{i=0}^{\infty} \frac{1}{2^i} = 2n = O(n)$$

Vagyis a Növel művelet amortizált költsége $O(n)/n = O(1)$

- Különböző műveletekre különböző költséget számolunk el
  - A i-edik műveletre elszámolt $\hat{c}_i$ *amortizációs költség* tetszőlegesen eltérhet annak c_i **tényleges költség**étől

- Különböző műveletekre különböző költséget számolunk el
  - A i-edik műveletre elszámolt $\hat{c}_i$ *amortizációs költség* tetszőlegesen eltérhet annak c_i **tényleges költség**étől
  - Azonban minden n hosszú műveletsorra teljesüljön, hogy

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

azaz a mindenkori hitelegyenleg $\left(\sum_{i=1}^{n} \hat{c}_i - c_i \right)$ nemnegatív

Könyvelési módszer használata – példa

- A NÖVEL művelet működése során könyveljünk el 2 egységnyi költséget egy bit 1-re állításához
- A költség fele a majdani visszaállításra félretett „hitel”

Könyvelési módszer használata – példa

- A NÖVEL művelet működése során könyveljünk el 2 egységnyi költséget egy bit 1-re állításához
- A költség fele a majdani visszaállításra félretett „hitel”
- A NÖVEL minden hívása során 1 bitet állítunk 1-re
  - n végrehajtás $\Rightarrow 2n$ összköltség $\Rightarrow O(1)$ költség/végrehajtás

Amortizált költségelemzés – Potenciálmódszer

- Az adatszerkezet i pillanatbeli állapotát jelöljük D_i-vel
- Vezessük be a $\Phi : \mathbb{N} \to \mathbb{R}$ **ponteciálfüggvény**t, ami az adatszerkezet egy D_i állapotához rendel egy **potenciál**t
- Az amortizációs költség legyen $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$

Amortizált költségelemzés – Potenciálmódszer

- Az adatszerkezet i pillanatbeli állapotát jelöljük D_i-vel
- Vezessük be a $\Phi : \mathbb{N} \to \mathbb{R}$ **ponteciálfüggvényt**, ami az adatszerkezet egy D_i állapotához rendel egy **potenciált**
- Az amortizációs költség legyen $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$

n hosszú műveletsorra a teljes *teleszkopikus összeg*

$$\sum_{i=1}^{n} \hat{c}_i = \Phi(D_n) - \Phi(D_0) + \sum_{i=1}^{n} c_i$$

Amortizált költségelemzés – Potenciálmódszer

- Az adatszerkezet i pillanatbeli állapotát jelöljük D_i-vel
- Vezessük be a $\Phi : \mathbb{N} \to \mathbb{R}$ **ponteciálfüggvényt**, ami az adatszerkezet egy D_i állapotához rendel egy **potenciált**
- Az amortizációs költség legyen $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$

n hosszú műveletsorra a teljes *teleszkopikus összeg*

$$\sum_{i=1}^{n} \hat{c}_i = \Phi(D_n) - \Phi(D_0) + \sum_{i=1}^{n} c_i$$

- Olyan potenciálfüggvényt keresünk, melyre $\Phi(D_n) \geq \Phi(D_0)$, mivel ekkor nyilvánvalóan $\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} c_i$ is teljesül

Amortizált költségelemzés – Potenciálmódszer

- Az adatszerkezet i pillanatbeli állapotát jelöljük D_i-vel
- Vezessük be a $\Phi : \mathbb{N} \to \mathbb{R}$ **ponteciálfüggvényt**, ami az adatszerkezet egy D_i állapotához rendel egy **potenciált**
- Az amortizációs költség legyen $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$

n hosszú műveletsorra a teljes *teleszkopikus összeg*

$$\sum_{i=1}^{n} \hat{c}_i = \Phi(D_n) - \Phi(D_0) + \sum_{i=1}^{n} c_i$$

- Olyan potenciálfüggvényt keresünk, melyre $\Phi(D_n) \geq \Phi(D_0)$, mivel ekkor nyilvánvalóan $\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} c_i$ is teljesül

Kényelmes megoldás

$\Phi(D_0) = 0$, és lássuk be, hogy $\Phi(D_i) \geq 0$

Amortizált költségelemzés – Példa

- Legyen $\Phi(D_i) = b_i$ a NÖVEL művelet i-szeri alkalmazására a számlálóban szereplő 1 értékű bitek száma
- Jelölje t_i a NÖVEL művelet i-edik végrehajtásakor 1-ről 0-ra változó bitek számát (vagyis $c_i \leq t_i + 1$)
  - $b_i \leq b_{i-1} - t_i + 1 \Rightarrow \Phi(D_i) - \Phi(D_{i-1}) \leq 1 - t_i$
  - Így az amortizációs költség $\hat{c}_i \leq c_i + 1 - t_i = t_i + 1 + 1 - t_i = 2$

Észrevétel

Mivel $\Phi(D_0) = 0$ és minden $\Phi(D_i) \geq 0$, így az n művelet amortizált költségének összege felső korlátja a tényleges összköltségnek ($O(n)$)