طراحی و تحلیل الگوریتم‌ها

مبحث هشتم

مباحث پیشرفته در طراحی و تحلیل الگوریتم‌ها

تحلیل سرشکنی

Amortized Analysis

کاظم فولادی
دانشکده مهندسی برق و کامپیوتر
دانشگاه تهران

http://courses.fouladi.ir/algorithm

جدول‌های پویا

DYNAMIC TABLES

می‌خواهیم جدول تا حد ممکن **کوچک** باشد (برای جلوگیری از اتلاف فضا)،
اما به اندازه‌ی کافی **بزرگ** باشد (برای جلوگیری از سرریز).

اما اندازه‌ی جدول از قبل مشخص نیست!
راه حل:
استفاده از **جدول‌های پویا**

ایده (جدول پویا):
هرگاه در جدول سرریز اتفاق افتاد،
- یک فضای جدید برای جدول تخصیص می‌دهیم (۲ برابر جدول قبلی)؛
- همه‌ی عناصر را از جدول قدیمی به جدول جدید منتقل می‌کنیم؛ و
- فضای جدول قدیمی را آزاد می‌کنیم.

جدول‌های پویا

شبه کد: درج در جدول پویا

DYNAMIC TABLES

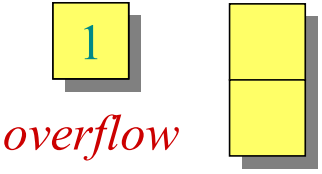
```
TABLE-INSERT(T, x)
 1  if T.size == 0
 2       allocate T.table with 1 slot
 3       T.size = 1
 4  if T.num == T.size
 5       allocate new-table with 2 · T.size slots
 6       insert all items in T.table into new-table
 7       free T.table
 8       T.table = new-table
 9       T.size = 2 · T.size
10  insert x into T.table
11  T.num = T.num + 1
```

جدول‌های پویا

مثال (۱ از ۱۱)

DYNAMIC TABLES

1. INSERT
2. INSERT

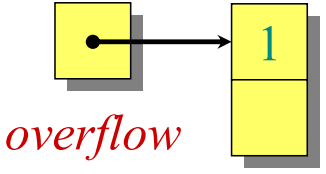


جدول‌های پویا

مثال (۲ از ۱۱)

DYNAMIC TABLES

1. INSERT
2. INSERT

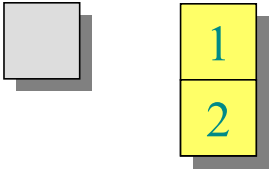


جدول‌های پویا

مثال (۳ از ۱۱)

DYNAMIC TABLES

1. INSERT
2. INSERT

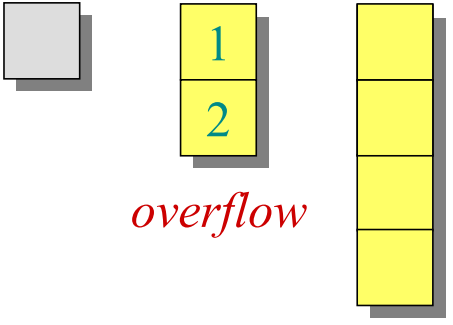


جدول‌های پویا

مثال (۴ از ۱۱)

DYNAMIC TABLES

1. INSERT
2. INSERT
3. INSERT

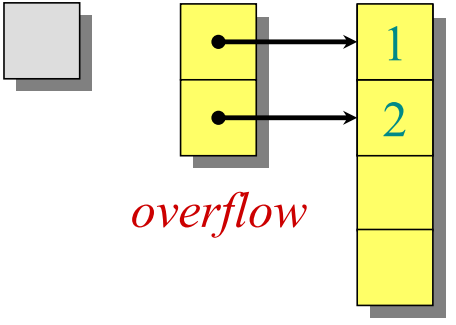


جدول‌های پویا

مثال (۵ از ۱۱)

DYNAMIC TABLES

1. INSERT
2. INSERT
3. INSERT

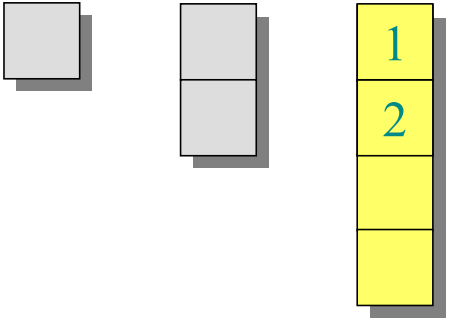


جدول‌های پویا

مثال (۶ از ۱۱)

DYNAMIC TABLES

1. INSERT
2. INSERT
3. INSERT

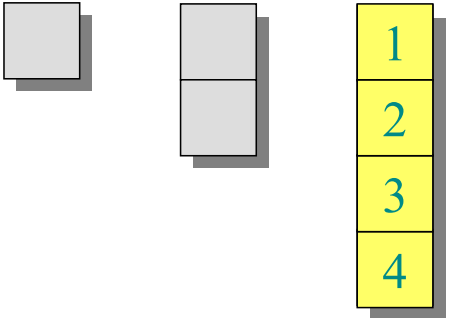


جدول‌های پویا

مثال (۷ از ۱۱)

DYNAMIC TABLES

1. INSERT
2. INSERT
3. INSERT
4. INSERT

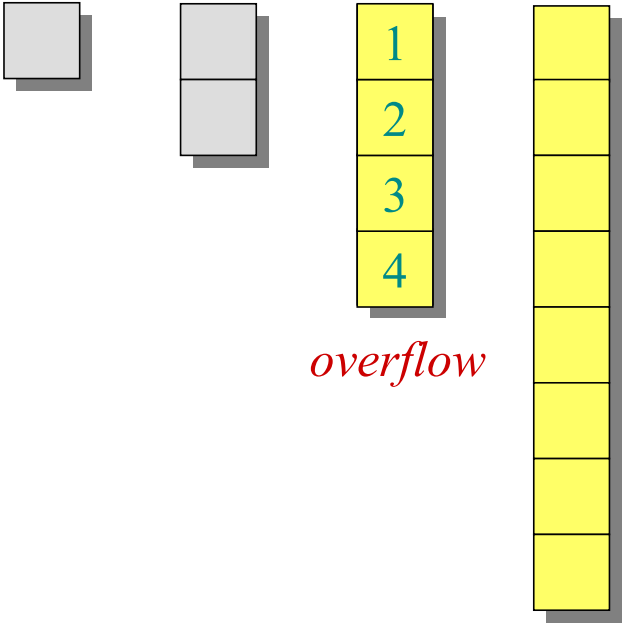


جدول‌های پویا

مثال (۸ از ۱۱)

DYNAMIC TABLES

1. INSERT
2. INSERT
3. INSERT
4. INSERT
5. INSERT

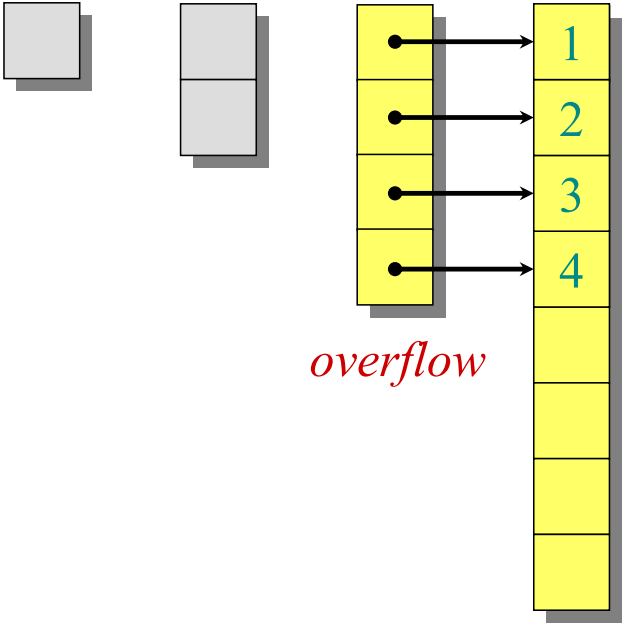


جدول‌های پویا

مثال (۹ از ۱۱)

DYNAMIC TABLES

1. INSERT
2. INSERT
3. INSERT
4. INSERT
5. INSERT

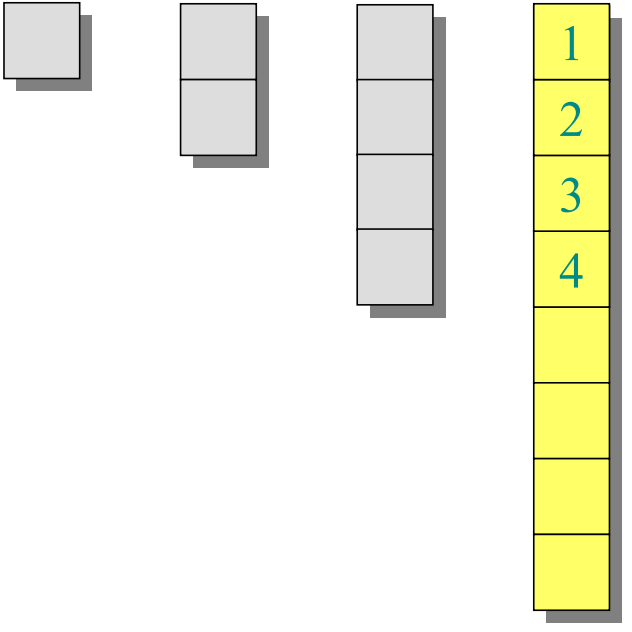


جدول‌های پویا

مثال (۱۰ از ۱۱)

DYNAMIC TABLES

1. INSERT
2. INSERT
3. INSERT
4. INSERT
5. INSERT

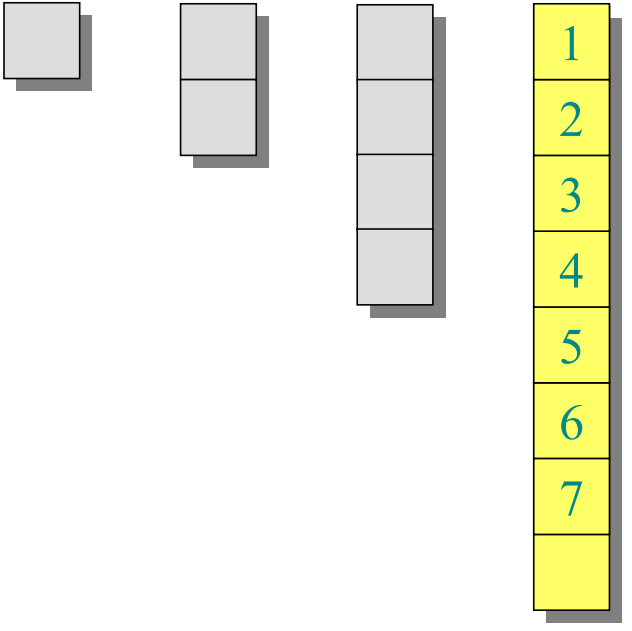


جدول‌های پویا

مثال (۱۱ از ۱۱)

DYNAMIC TABLES

1. INSERT
2. INSERT
3. INSERT
4. INSERT
5. INSERT
6. INSERT
7. INSERT



جدول‌های پویا

تحلیل زمان اجرا در بدترین حالت

یک دنباله از n عمل درج را در نظر می‌گیریم:

زمان اجرای بدترین حالت برای یک درج $\Theta(n)$ است.
پس: زمان اجرا برای n عمل درج می‌شود:

$$n \cdot \Theta(n) = \Theta(n^2)$$

اما این نتیجه اشتباه است!!

در واقع، زمان اجرای بدترین حالت برای یک دنباله از n عمل درج تنها عبارت است از:

$$\Theta(n) \ll \Theta(n^2)$$

چرا؟

جدول‌های پویا

تحلیل دقیق‌تر زمان اجرا (۱ از ۳)

تعریف می‌کنیم: c_i = هزینه‌ی i–امین درج

$$c_i = \begin{cases} i & , i-1 = 2^k, k \in \mathbb{Z}^{\geq 0} \\ 1 & , \text{otherwise} \end{cases}$$

i	1	2	3	4	5	6	7	8	9	10
$size_i$	1	2	4	4	8	8	8	8	16	16
c_i	1	2	3	1	5	1	1	1	9	1

جدول‌های پویا

تحلیل دقیق‌تر زمان اجرا (۲ از ۳)

تعریف می‌کنیم: c_i = هزینه‌ی i–امین درج

$$c_i = \begin{cases} i & , i-1 = 2^k, k \in \mathbb{Z}^{\geq 0} \\ 1 & , \text{otherwise} \end{cases}$$

i	1	2	3	4	5	6	7	8	9	10
$size_i$	1	2	4	4	8	8	8	8	16	16
c_i	1	1 + 1	1 + 2	1	1 + 4	1	1	1	1 + 8	1

جدول‌های پویا

تحلیل دقیق‌تر زمان اجرا (۳ از ۳)

تعریف می‌کنیم: c_i = هزینه‌ی i–امین درج

$$c_i = \begin{cases} i & , i-1 = 2^k, k \in \mathbb{Z}^{\geq 0} \\ 1 & , \text{otherwise} \end{cases}$$

هزینه‌ی n عمل درج می‌شود:

$$\sum_{i=1}^{n} c_i \leq n + \sum_{j=0}^{\lfloor \log_2(n-1) \rfloor} 2^j \leq 3n = \Theta(n)$$

پس هزینه‌ی متوسط هر عمل درج در جدول پویا می‌شود:

$$\Theta(n) \;/\; n = \Theta(1)$$

تحلیل سرشکنی

AMORTIZED ANALYSIS

تحلیل سرشکنی

راهبردی است که یک دنباله از عمل‌ها را تحلیل می‌کند
تا نشان بدهد هزینه‌ی متوسط هر عمل کوچک است؛
حتی اگر یک عمل واحد در آن دنباله گران باشد.

اگرچه از میانگین استفاده می‌کنیم، اما احتمالات در اینجا درگیر نمی‌شوند.

تحلیل سرشکنی، کارآیی متوسط هر عمل را در بدترین حالت تضمین می‌کند.

هزینه‌ی سرشکن‌شده

AMORTIZED COST

هزینه‌ی سرشکن‌شده

یک ساختمان داده و دنباله‌ای از عملیات مختلف بر روی آن را در نظر می‌گیریم.

هزینه‌ی سرشکن‌شده‌ی هر عملیات برابر با
میانگین هزینه‌ی بدترین حالت آن عملیات در این دنباله تعریف می‌شود.

انواع تحلیل سرشکنی

TYPES OF AMORTIZED ANALYSES

سه روش تحلیل سرشکنی متداول		
روش تجمعی *Aggregate Method*	**روش حسابداری** *Accounting Method*	**روش پتانسیل** *Potential Method*

انواع تحلیل سرشکنی

روش تجمعی

AGGREGATE METHOD

سه روش تحلیل سرشکنی متداول		
روش تجمعی *Aggregate Method*	روش حسابداری *Accounting Method*	روش پتانسیل *Potential Method*
ساده، کمتر بودن دقت نسبت به دو روش دیگر (محاسبه‌ی مجموع هزینه‌ها تقسیم بر تعداد عملیات)		

انواع تحلیل سرشکنی

روش حسابداری

ACCOUNTING METHOD

سه روش تحلیل سرشکنی متداول

روش پتانسیل
Potential Method

روش حسابداری
Accounting Method

روش تجمعی
Aggregate Method

- هر عملیات با یک هزینه‌ی سرشکن‌شده‌ی فرضی **شارژ** می‌شود.
- این مبلغ برای انجام آن عملیات مصرف می‌شود.
- هر مبلغی که بلافاصله مصرف نشود، برای استفاده توسط عملیات بعدی در **مخزن** ذخیره می‌شود.

روش حسابداری

ACCOUNTING METHOD

- هر عملیات با یک هزینه‌ی سرشکن‌شده‌ی فرضی **شارژ** می‌شود.
  هزینه‌ی سرشکن‌شده‌ی فرضی عملیات i–ام: $\hat{c}_i$
- از این مبلغ برای انجام آن عملیات مصرف می‌شود.
- هر مبلغی که بلافاصله مصرف نشود، برای استفاده توسط عملیات بعدی در **مخزن** (بانک) ذخیره می‌شود.
- توازن بانکی نباید منفی شود ⇐ باید مطمئن باشیم که برای هر n

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

- در این صورت، مجموع هزینه‌های سرشکن‌شده،
  یک کران بالا بر روی مجموع هزینه‌های واقعی فراهم می‌کند.

روش حسابداری

مثال: تحلیل سرشکنی زمان اجرا با روش حسابداری (۱ از ۴)

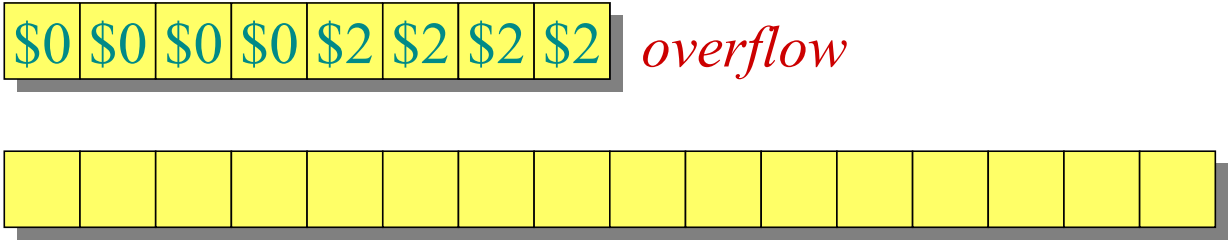
ACCOUNTING ANALYSIS OF DYNAMIC TABLES

شارژ هزینه‌ی سرشکن‌شده‌ی فرضی عملیات درج i–ام با $\hat{c}_i = \$3$

- $1 هزینه‌ی پرداختی برای درج فوری
- $2 ذخیره می‌شود برای دوبرابر شدن اندازه‌ی جدول

(وقتی جدول دو برابر می‌شود،
$1 برای انتقال عنصر جدید و $1 برای انتقال عنصر قدیمی صرف می‌شود.)

مثال



روش حسابداری

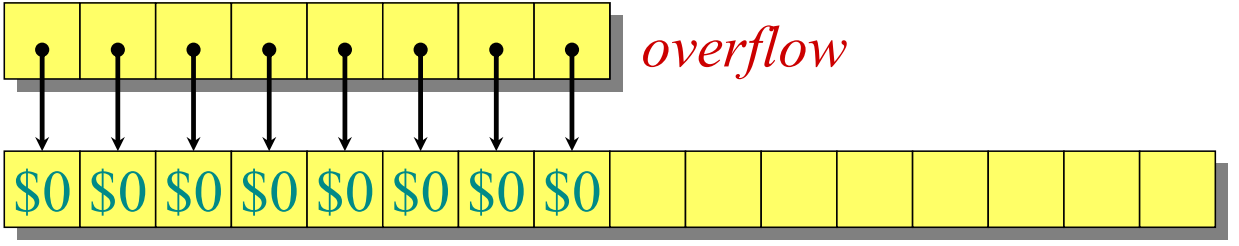
مثال: تحلیل سرشکنی زمان اجرا با روش حسابداری (۲ از ۴)

ACCOUNTING ANALYSIS OF DYNAMIC TABLES

شارژ هزینه‌ی سرشکن‌شده‌ی فرضی عملیات درج i–ام با $\hat{c}_i = \$3$
- $\$1$ هزینه‌ی پرداختی برای درج فوری
- $\$2$ ذخیره می‌شود برای دوبرابر شدن اندازه‌ی جدول

(وقتی جدول دو برابر می‌شود،
$\$1$ برای انتقال عنصر جدید و $\$1$ برای انتقال عنصر قدیمی صرف می‌شود.)

مثال



روش حسابداری

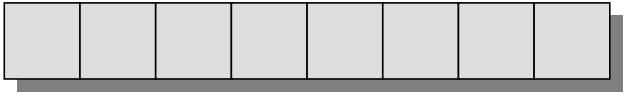
مثال: تحلیل سرشکنی زمان اجرا با روش حسابداری (۳ از ۴)

ACCOUNTING ANALYSIS OF DYNAMIC TABLES

شارژ هزینه‌ی سرشکن‌شده‌ی فرضی عملیات درج i–ام با $\hat{c}_i = \$3$
- $\$1$ هزینه‌ی پرداختی برای درج فوری
- $\$2$ ذخیره می‌شود برای دوبرابر شدن اندازه‌ی جدول

(وقتی جدول دو برابر می‌شود،
$\$1$ برای انتقال عنصر جدید و $\$1$ برای انتقال عنصر قدیمی صرف می‌شود.)

مثال



روش حسابداری

مثال: تحلیل سرشکنی زمان اجرا با روش حسابداری (۴ از ۴)

ACCOUNTING ANALYSIS OF DYNAMIC TABLES

شارژ هزینه‌ی سرشکن‌شده‌ی فرضی عملیات درج i–ام با $\hat{c}_i = \$3$

- $1 هزینه‌ی پرداختی برای درج فوری
- $2 ذخیره می‌شود برای دوبرابر شدن اندازه‌ی جدول

(وقتی جدول دو برابر می‌شود،
$1 برای انتقال عنصر جدید و $1 برای انتقال عنصر قدیمی صرف می‌شود.)

i	1	2	3	4	5	6	7	8	9	10
$size_i$	1	2	4	4	8	8	8	8	16	16
c_i	1	2	3	1	5	1	1	1	9	1
$\hat{c}_i$	2*	3	3	3	3	3	3	3	3	3
$bank_i$	1	2	2	4	2	4	6	8	2	4

* The first operation costs only $2, not $3.

توازن بانکی هرگز به زیرصفر نمی‌رود.

پس، مجموع هزینه‌های سرشکن‌شده، یک کران بالا بر روی مجموع هزینه‌های واقعی فراهم می‌کند.

انواع تحلیل سرشکنی

روش پتانسیل

POTENTIAL METHOD

سه روش تحلیل سرشکنی متداول

روش پتانسیل *Potential Method*	روش حسابداری *Accounting Method*	روش تجمعی *Aggregate Method*

به حساب بانکی به‌عنوان
انرژی پتانسیل
یک مجموعه‌ی پویا
نگاه می‌کنیم.

روش پتانسیل

POTENTIAL METHOD

به حساب بانکی به‌عنوان انرژی پتانسیل یک مجموعه‌ی پویا نگاه می‌کنیم.

- با یک ساختمان‌داده‌ی اولیه D_0 شروع می‌کنیم.
- عملیات i–ام، D_{i-1} را به D_i تبدیل می‌کند.
- هزینه‌ی عملیات i–ام، c_i است.
- یک **تابع پتانسیل** Φ را به صورت زیر تعریف می‌کنیم:

$$\Phi : \{D_i\} \to \mathbb{R}, \quad \Phi(D_0) = 0, \quad \Phi(D_i) \geq 0 \quad \forall i$$

- هزینه‌ی سرشکن‌شده‌ی $\hat{c}_i$ با توجه به تابع پتانسیل Φ تعریف می‌شود:

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

روش پتانسیل

مفهوم تابع پتانسیل

POTENTIAL FUNCTION

$$\hat{c}_i = c_i + \underbrace{\Phi(D_i) - \Phi(D_{i-1})}_{\textbf{\textit{potential difference}}\ \Delta\Phi_i}$$

$$\text{If } \Delta\Phi_i > 0, \text{ then } \hat{c}_i > c_i$$

عملیات i–ام کار را در ساختمان داده ذخیره می‌کند برای استفاده‌ی بعدی.

$$\text{If } \Delta\Phi_i < 0, \text{ then } \hat{c}_i < c_i$$

ساختمان داده کار ذخیره شده را برای کمک به انجام عملیات i–ام تحویل می‌دهد.

روش پتانسیل

مجموع هزینه‌های سرشکن‌شده کران پایین مجموع هزینه‌های واقعی می‌شود

THE AMORTIZED COSTS BOUND THE TRUE COSTS

مجموع هزینه‌ی سرشکن‌شده‌ی یک دنباله از n عملیات می‌شود:

$$\sum_{i=1}^{n} \hat{c}_i = \sum_{i=1}^{n} \left(c_i + \Phi(D_i) - \Phi(D_{i-1})\right)$$

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

$$\geq \sum_{i=1}^{n} c_i \qquad \text{since } \Phi(D_n) \geq 0 \text{ and } \Phi(D_0) = 0.$$

روش پتانسیل

مثال: تحلیل سرشکنی زمان اجرا با روش پتانسیل (۱ از ۴)

POTENTIAL ANALYSIS OF DYNAMIC TABLES

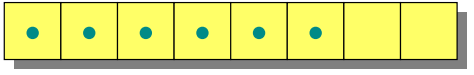
تعریف پتانسیل جدول پس از i–امین درج (با فرض $2^{\lceil \lg 0 \rceil} = 0$)

$$\Phi(D_i) = 2i - 2^{\lceil \lg i \rceil}$$

Note:

- $\Phi(D_0) = 0$,
- $\Phi(D_i) \geq 0$ for all i.

مثال



$\Phi = 2 \cdot 6 - 2^3 = 4$

(

accounting method)

روش پتانسیل

مثال: تحلیل سرشکنی زمان اجرا با روش پتانسیل (۲ از ۴)

POTENTIAL ANALYSIS OF DYNAMIC TABLES

هزینه‌ی سرشکن‌شده‌ی i–امین درج

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

$$= \begin{cases} i + \left(2i - 2^{\lceil \lg i \rceil}\right) - \left(2(i-1) - 2^{\lceil \lg (i-1) \rceil}\right) \\ \quad \text{if } i-1 \text{ is an exact power of } 2, \\ 1 + \left(2i - 2^{\lceil \lg i \rceil}\right) - \left(2(i-1) - 2^{\lceil \lg (i-1) \rceil}\right) \\ \quad \text{otherwise.} \end{cases}$$

$$\Phi(D_i) = 2i - 2^{\lceil \lg i \rceil}$$

روش پتانسیل

مثال: تحلیل سرشکنی زمان اجرا با روش پتانسیل (۳ از ۴)

POTENTIAL ANALYSIS OF DYNAMIC TABLES

هزینه‌ی سرشکن‌شده‌ی i–امین درج

Case 1: $i-1$ is an exact power of 2.

$$\begin{aligned}\hat{c}_i &= i + \left(2i - 2^{\lceil \lg i \rceil}\right) - \left(2(i-1) - 2^{\lceil \lg (i-1) \rceil}\right) \\ &= i + 2 - \left(2^{\lceil \lg i \rceil} - 2^{\lceil \lg (i-1) \rceil}\right) \\ &= i + 2 - \left(2(i-1) - (i-1)\right) \\ &= i + 2 - 2i + 2 + i - 1 \\ &= 3\end{aligned}$$

روش پتانسیل

مثال: تحلیل سرشکنی زمان اجرا با روش پتانسیل (۴ از ۴)

POTENTIAL ANALYSIS OF DYNAMIC TABLES

هزینه‌ی سرشکن‌شده‌ی i–امین درج

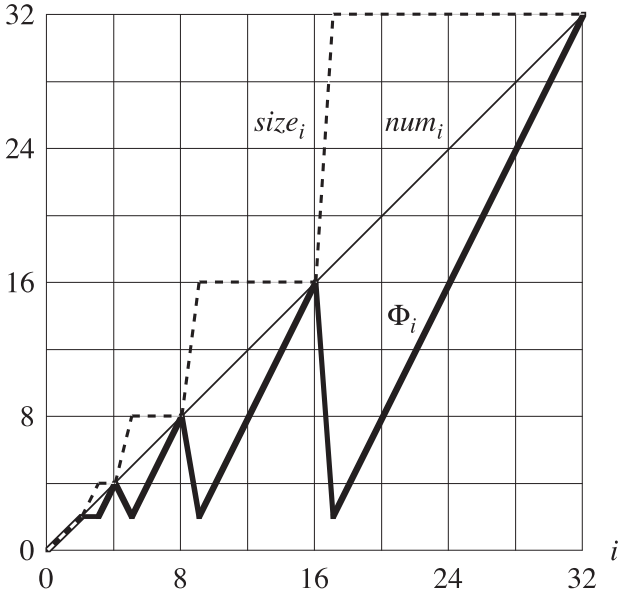
Case 2: $i - 1$ is not an exact power of 2.

$$\begin{aligned}\hat{c}_i &= 1 + \left(2i - 2^{\lceil \lg i \rceil}\right) - \left(2(i-1) - 2^{\lceil \lg (i-1) \rceil}\right) \\ &= 1 + 2 - \left(2^{\lceil \lg i \rceil} - 2^{\lceil \lg (i-1) \rceil}\right) \\ &= 3\end{aligned}$$

Therefore, n insertions cost $\Theta(n)$ in the worst case.

روش پتانسیل

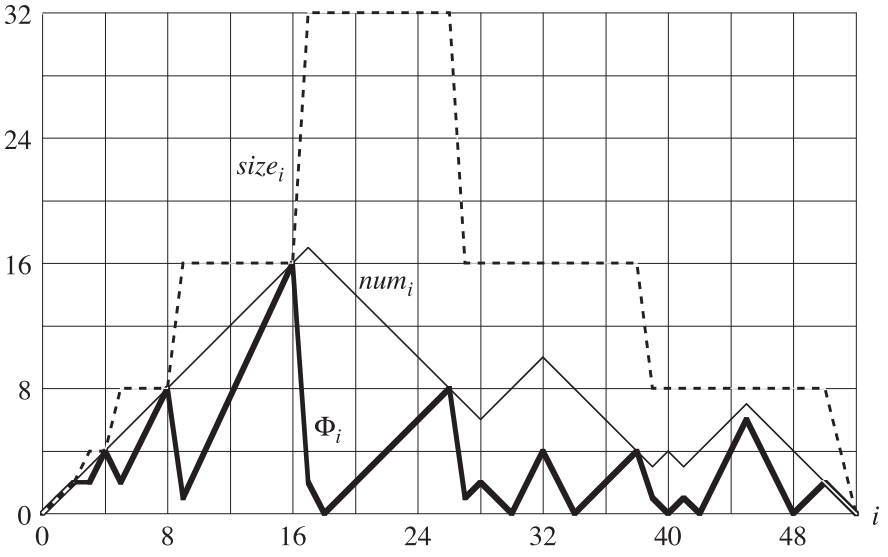
تحلیل سرشکنی زمان اجرا با روش پتانسیل



نمودارهای اندازه، تعداد عناصر و تابع پتانسیل به‌ازای درج‌های متوالی در یک جدول پویا
(در حالتی که عمل حذف نداریم)

روش پتانسیل

تحلیل سرشکنی زمان اجرا با روش پتانسیل



نمودارهای اندازه، تعداد عناصر و تابع پتانسیل به‌ازای درج‌ها و حذف‌های متوالی در یک جدول پویا

مثال: پشته با عملیات چندگانه

فرض می‌کنیم بر روی یک پشته عملیات زیر تعریف شده باشد:

PUSH	POP	MULTIPOP
		پارامتر k را می‌گیرد و k عنصر بالای پشته را یک به یک حذف می‌کند. (اگر تعداد عناصر موجود در پشته کمتر از k بود، همه‌ی عناصر یک به یک حذف می‌شوند.)

```
MULTIPOP(S, k)
1  while not STACK-EMPTY(S) and k > 0
2      POP(S)
3      k = k − 1
```

مثال: پشته با عملیات چندگانه

مثال از MULTIPOP

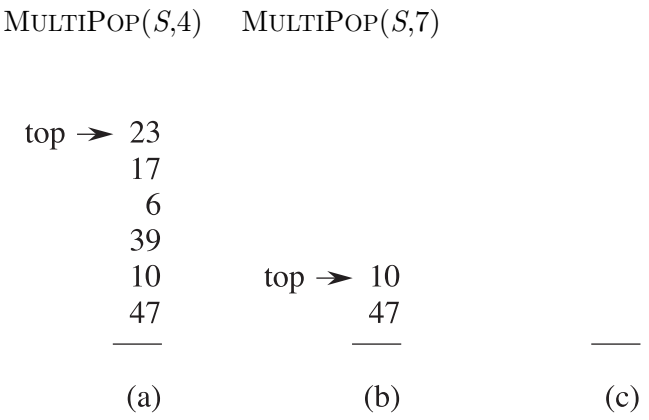


Figure 17.1 The action of MULTIPOP on a stack S, shown initially in **(a)**. The top 4 objects are popped by MULTIPOP$(S, 4)$, whose result is shown in **(b)**. The next operation is MULTIPOP$(S, 7)$, which empties the stack—shown in **(c)**—since there were fewer than 7 objects remaining.

مثال: پشته با عملیات چندگانه

زمان اجرای هر عملیات

PUSH	POP	MULTIPOP
PUSH(S,x)	POP(S,x)	MULTIPOP(S,k)
$O(1)$	$O(1)$	$O(\min\{s,k\})$ $s = \text{size}(S)$

تحلیل سرشکنی: یک دنباله از n عملیات فوق بر روی یک پشته‌ی خالی را در نظر می‌گیریم.
در بدترین حالت مجموع زمان مصرف شده برای این دنباله عملیات چیست؟

پرهزینه‌ترین عملیات در این بین MULTIPOP(S,n) با هزینه‌ی $O(n)$ است،
پس هزینه‌ی دنباله‌ای از n عملیات می‌شود $n \times O(n) = O(n^2)$.
این نتیجه درست، اما نادقیق است!!

مثال: پشته با عملیات چندگانه

تحلیل سرشکنی با روش تحلیل تجمعی

PUSH	POP	MULTIPOP
$\text{PUSH}(S,x)$	$\text{POP}(S,x)$	$\text{MULTIPOP}(S,k)$

تحلیل سرشکنی: یک دنباله از n عملیات فوق بر روی یک پشته‌ی خالی را در نظر می‌گیریم.
در بدترین حالت مجموع زمان مصرف شده برای این دنباله عملیات چیست؟

اگر n عمل داشته باشیم، حداکثر تعداد عناصر پشته می‌شود n (وقتی همه PUSH باشند.)
هر عنصر داخل پشته، دقیقاً یک‌بار PUSH و حداکثر یک بار POP می‌شود.
یک عنصر یا مستقیماً با عمل POP حذف می‌شود یا با MULTIPOP.

پس در مجموع حداکثر $2n$ بار عمل PUSH و POP عادی انجام می‌شود که هزینه‌ی هر یک $O(1)$ است.
پس هزینه‌ی سرشکن‌شده‌ی هر عمل متناسب است با

$$2n \;/\; n = 2 = O(1)$$

مثال: پشته با عملیات چندگانه

تحلیل سرشکنی با روش حساب‌داری

PUSH	POP	MULTIPOP
PUSH(S,x)	POP(S,x)	MULTIPOP(S,k)
هزینه‌ی واقعی = 1	هزینه‌ی واقعی = 1	هزینه‌ی واقعی = 1
هزینه‌ی سرشکن = 2	هزینه‌ی سرشکن = 0	هزینه‌ی سرشکن = $\min\{s,k\}$

تحلیل سرشکنی: یک دنباله از n عملیات فوق بر روی یک پشته‌ی خالی را در نظر می‌گیریم.
در بدترین حالت مجموع زمان مصرف شده برای این دنباله عملیات چیست؟

اگر n عمل داشته باشیم، در بدترین حالت همه‌ی این عملیات PUSH است و با توجه به

$$\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} c_i = 2n$$

مقدار $2n$ کران بالا برای n عمل PUSH است.

بنابراین، برای هر دنباله از n عمل سه‌گانه‌ی فوق، هزینه‌ی سرشکن‌شده‌ی کل، یک کران بالا برای هزینه‌ی واقعی کل است. چون هزینه‌ی سرشکن‌شده‌ی کل $O(n)$ است، پس هزینه‌ی سرشکن‌شده‌ی هر عمل می‌شود:

$$O(n) \;/\; n = O(1)$$

مثال: پشته با عملیات چندگانه

تحلیل سرشکنی با روش پتانسیل

PUSH	POP	MULTIPOP
$\text{PUSH}(S,x)$	$\text{POP}(S,x)$	$\text{MULTIPOP}(S,k)$

تحلیل سرشکنی: یک دنباله از n عملیات فوق بر روی یک پشته‌ی خالی را در نظر می‌گیریم.
در بدترین حالت مجموع زمان مصرف شده برای این دنباله عملیات چیست؟

$$\Phi(D_i)$$

تابع پتانسیل را تعداد عناصر موجود در پشته در نظر می‌گیریم.

در ابتدا پشته خالی است، پس:

$$\Phi(D_0) = 0$$

در نتیجه

$$\Phi(D_n) \geq \Phi(D_0) = 0$$

مثال: پشته با عملیات چندگانه

تحلیل سرشکنی با روش پتانسیل: عمل PUSH

PUSH	POP	MULTIPOP
PUSH(S,x)	POP(S,x)	MULTIPOP(S,k)

تحلیل سرشکنی: یک دنباله از n عملیات فوق بر روی یک پشته‌ی خالی را در نظر می‌گیریم.
در بدترین حالت مجموع زمان مصرف شده برای این دنباله عملیات چیست؟

اگر عمل iام عمل PUSH باشد، چون یک عنصر اضافه می‌شود، داریم:

$$\Phi(D_i) - \Phi(D_{i-1}) = 1$$

هزینه‌ی واقعی این عمل $c_i = 1$ است، پس:

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1}) = 1 + 1 = 2$$

مثال: پشته با عملیات چندگانه

تحلیل سرشکنی با روش پتانسیل: عمل POP

PUSH
PUSH(S,x)

POP
POP(S,x)

MULTIPOP
MULTIPOP(S,k)

تحلیل سرشکنی: یک دنباله از n عملیات فوق بر روی یک پشته‌ی خالی را در نظر می‌گیریم.
در بدترین حالت مجموع زمان مصرف شده برای این دنباله عملیات چیست؟

اگر عمل iام عمل POP باشد، چون یک عنصر کم می‌شود، داریم:

$$\Phi(D_i) - \Phi(D_{i-1}) = -1$$

هزینه‌ی واقعی این عمل $c_i = 1$ است، پس:

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1}) = 1 + (-1) = 0$$

مثال: پشته با عملیات چندگانه

تحلیل سرشکنی با روش پتانسیل: عمل MULTIPOP

PUSH
PUSH(S,x)

POP
POP(S,x)

MULTIPOP
MULTIPOP(S,k)

تحلیل سرشکنی: یک دنباله از n عملیات فوق بر روی یک پشته‌ی خالی را در نظر می‌گیریم.
در بدترین حالت مجموع زمان مصرف شده برای این دنباله عملیات چیست؟

اگر عمل iام عمل MILTIPOP باشد، چون در واقع تعدادی POP است،
پس هزینه‌ی سرشکن‌شده‌ی آن هم صفر می‌شود.

$$\hat{c}_i = 0$$

مثال: پشته با عملیات چندگانه

تحلیل سرشکنی با روش پتانسیل: نتیجه

PUSH	POP	MULTIPOP
PUSH(S,x)	POP(S,x)	MULTIPOP(S,k)

تحلیل سرشکنی: یک دنباله از n عملیات فوق بر روی یک پشته‌ی خالی را در نظر می‌گیریم.
در بدترین حالت مجموع زمان مصرف شده برای این دنباله عملیات چیست؟

چون تابع پتانسیل صعودی است،
هزینه‌ی سرشکن‌شده‌ی کل یک دنباله از n عملیات یک کران بالا برای هزینه‌ی واقعی کل است. داریم:

$$\sum_{i=1}^{n} c_i \leq \sum_{i=1}^{n} \hat{c}_i = 2n$$

پس، هزینه‌ی سرشکن‌شده‌ی هر عمل می‌شود:

$$O(n) \;/\; n = O(1)$$

شمارنده‌ی دودویی

یک شمارنده‌ی دودویی k بیتی $A[k-1..0]$ را در نظر می‌گیریم.
مقدار اولیه‌ی آن صفر است.
تنها عمل تعریف شده بر روی آن «افزایش یک واحد» به شمارنده است.

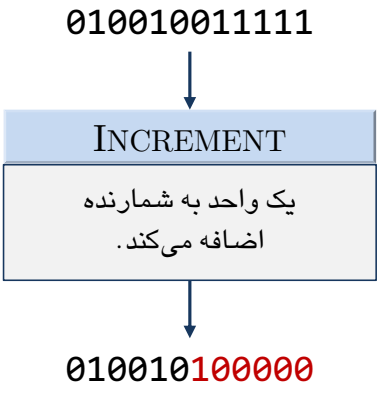
INCREMENT
یک واحد به شمارنده اضافه می‌کند.

```
INCREMENT(A)
1  i = 0
2  while i < A.length and A[i] == 1
3       A[i] = 0
4       i = i + 1
5  if i < A.length
6       A[i] = 1
```

شمارنده‌ی دودویی

مثال

```
INCREMENT(A)
1  i = 0
2  while i < A.length and A[i] == 1
3        A[i] = 0
4        i = i + 1
5  if i < A.length
6        A[i] = 1
```



۶ تغییر بیت:
هزینه‌ی افزایش = ۶

شمارنده‌ی دودویی

مثال

Counter value	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]	Total cost
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1
2	0	0	0	0	0	0	1	0	3
3	0	0	0	0	0	0	1	1	4
4	0	0	0	0	0	1	0	0	7
5	0	0	0	0	0	1	0	1	8
6	0	0	0	0	0	1	1	0	10
7	0	0	0	0	0	1	1	1	11
8	0	0	0	0	1	0	0	0	15
9	0	0	0	0	1	0	0	1	16
10	0	0	0	0	1	0	1	0	18
11	0	0	0	0	1	0	1	1	19
12	0	0	0	0	1	1	0	0	22
13	0	0	0	0	1	1	0	1	23
14	0	0	0	0	1	1	1	0	25
15	0	0	0	0	1	1	1	1	26
16	0	0	0	1	0	0	0	0	31

یک شمارنده که از مقدار 0 تا 16 شمارش می‌کند.

هزینه‌ی هر عمل «افزایش یک»:
بر اساس تعداد بیت های تغییریافته خطی است.

در بدترین حالت:
زمان اجرای هر «افزایش یک» برابر با $O(k)$ است.

پس: زمان اجرای بدترین حالت دنباله‌ای از n عملیات «افزایش یک» روی شمارنده‌ای با مقدار اولیه‌ی صفر، برابر است با $O(nk)$.

این نتیجه درست، اما نادقیق است!!

شمارنده‌ی دودویی

تحلیل سرشکنی با روش تحلیل تجمعی

تحلیل سرشکنی: یک دنباله از n عملیات «افزایش یک» را بر روی یک شمارنده‌ی دودویی در نظر می‌گیریم.
در بدترین حالت مجموع زمان مصرف شده برای این دنباله عملیات چیست؟

Counter value	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]	Total cost
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1
2	0	0	0	0	0	0	1	0	3
3	0	0	0	0	0	0	1	1	4
4	0	0	0	0	0	1	0	0	7
5	0	0	0	0	0	1	0	1	8
6	0	0	0	0	0	1	1	0	10
7	0	0	0	0	0	1	1	1	11
8	0	0	0	0	1	0	0	0	15
9	0	0	0	0	1	0	0	1	16
10	0	0	0	0	1	0	1	0	18
11	0	0	0	0	1	0	1	1	19
12	0	0	0	0	1	1	0	0	22
13	0	0	0	0	1	1	0	1	23
14	0	0	0	0	1	1	1	0	25
15	0	0	0	0	1	1	1	1	26
16	0	0	0	1	0	0	0	0	31

تعداد کل تغییر بیت‌ها را می‌شماریم:

- $A[0]$: با هر افزایش یک تغییر می‌کند، در مجموع: $n = \lfloor n/2^0 \rfloor$ تغییر
- $A[1]$: با هر ۲ بار افزایش یک تغییر می‌کند، در مجموع: $\lfloor n/2^1 \rfloor = n/2$ تغییر
- $A[2]$: با هر ۴ بار افزایش یک تغییر می‌کند، در مجموع: $\lfloor n/2^2 \rfloor = n/4$ تغییر
- …
- $A[k]$: با هر $n/2^k$ بار افزایش یک تغییر می‌کند، در مجموع: $\lfloor n/2^k \rfloor$ تغییر

پس تعداد تغییر بیت‌ها در مجموع می‌شود:

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

و هزینه‌ی سرشکن‌شده عبارت است از:

$$2n \;/\; n = 2 = O(1)$$

شمارنده‌ی دودویی

تحلیل سرشکنی با روش حسابداری

تحلیل سرشکنی: یک دنباله از n عملیات «افزایش یک» را بر روی یک شمارنده‌ی دودویی در نظر می‌گیریم.
در بدترین حالت مجموع زمان مصرف شده برای این دنباله عملیات چیست؟

Counter value	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]	Total cost
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1
2	0	0	0	0	0	0	1	0	3
3	0	0	0	0	0	0	1	1	4
4	0	0	0	0	0	1	0	0	7
5	0	0	0	0	0	1	0	1	8
6	0	0	0	0	0	1	1	0	10
7	0	0	0	0	0	1	1	1	11
8	0	0	0	0	1	0	0	0	15
9	0	0	0	0	1	0	0	1	16
10	0	0	0	0	1	0	1	0	18
11	0	0	0	0	1	0	1	1	19
12	0	0	0	0	1	1	0	0	22
13	0	0	0	0	1	1	0	1	23
14	0	0	0	0	1	1	1	0	25
15	0	0	0	0	1	1	1	1	26
16	0	0	0	1	0	0	0	0	31

برای هر عمل «افزایش یک»، ۲ واحد هزینه می‌کنیم.
حداکثر یک بیت از 0 به 1 تغییر می‌کند (اگر همه‌ی بیت‌ها 1 باشند، پس از افزایش همه 0 می‌شوند).
۱ واحد صرف تغییر می‌شود؛
۱ واحد باقی‌مانده در مخزن ذخیره می‌شود.

⇐ پس از مدتی، برای هر کدام از بیت‌های 1، ۱ واحد در مخزن قرار دارد.
⇐ پس عمل تغییر بیت از 1 به 0 مجانی انجام می‌شود.

⇐ اگر n بار عمل «افزایش یک» را انجام بدهیم، دقیقاً $2n$ واحد هزینه کرده‌ایم و ممکن است در انتها مقداری پول در مخزن اضافه بماند.

⇐ پس هزینه‌ی سرشکن‌شده می‌شود:

$$2n \;/\; n = 2 = O(1)$$

شمارنده‌ی دودویی

تحلیل سرشکنی با روش پتانسیل

تحلیل سرشکنی: یک دنباله از n عملیات «افزایش یک» را بر روی یک شمارنده‌ی دودویی در نظر می‌گیریم.
در بدترین حالت مجموع زمان مصرف شده برای این دنباله عملیات چیست؟

Counter value	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]	Total cost
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1
2	0	0	0	0	0	0	1	0	3
3	0	0	0	0	0	0	1	1	4
4	0	0	0	0	0	1	0	0	7
5	0	0	0	0	0	1	0	1	8
6	0	0	0	0	0	1	1	0	10
7	0	0	0	0	0	1	1	1	11
8	0	0	0	0	1	0	0	0	15
9	0	0	0	0	1	0	0	1	16
10	0	0	0	0	1	0	1	0	18
11	0	0	0	0	1	0	1	1	19
12	0	0	0	0	1	1	0	0	22
13	0	0	0	0	1	1	0	1	23
14	0	0	0	0	1	1	1	0	25
15	0	0	0	0	1	1	1	1	26
16	0	0	0	1	0	0	0	0	31

$$\Phi(D_i)$$

تابع پتانسیل را تعداد یک‌های موجود در آرایه در نظر می‌گیریم.
در ابتدا شمارنده صفر است (و هیچ یکی ندارد)، پس:

$$\Phi(D_0) = 0$$

در نتیجه

$$\Phi(D_n) \geq \Phi(D_0) = 0$$

یک عمل «افزایش یک» را در نظر می‌گیریم.
به‌فرض: در این عمل t_i بیت از 1 به 0 و حداکثر ۱ بیت از 0 به 1 تغییر می‌کند، پس

$$\left.\begin{aligned} \hat{c}_i &= \Phi(D_i) - \Phi(D_{i-1}) + c_i \\ \Phi(D_i) - \Phi(D_{i-1}) &\leq -t_i + 1 \\ c_i &\leq t_i + 1 \end{aligned}\right\} \Rightarrow \hat{c}_i \leq 2$$

پس هزینه‌ی سرشکن‌شده‌ی هر عمل «افزایش یک» می‌شود:

$$2n \;/\; n = 2 = O(1)$$