بسم الله الرحمن الرحیم

ساختمان‌های داده

جلسه ۱۰

مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

تحلیل سرشکنی (Amortized)

- تاکنون با تحلیل الگوریتم‌ها در بدترین حالت، حالت میانگین و بهترین حالت آشنا شدیم. اما گاهی با **دنباله‌ای از اعمال** بر روی داده ساختاری سر و کار داریم که هزینه برخی از آنها در مواردی خیلی زیاد اما تعداد رخداد این موارد به نسبت کم است.

تحلیل سرشکنی (Amortized)

- یک داده ساختار و دنباله‌ای از اعمال مختلف بر روی آن را در نظر بگیرید.

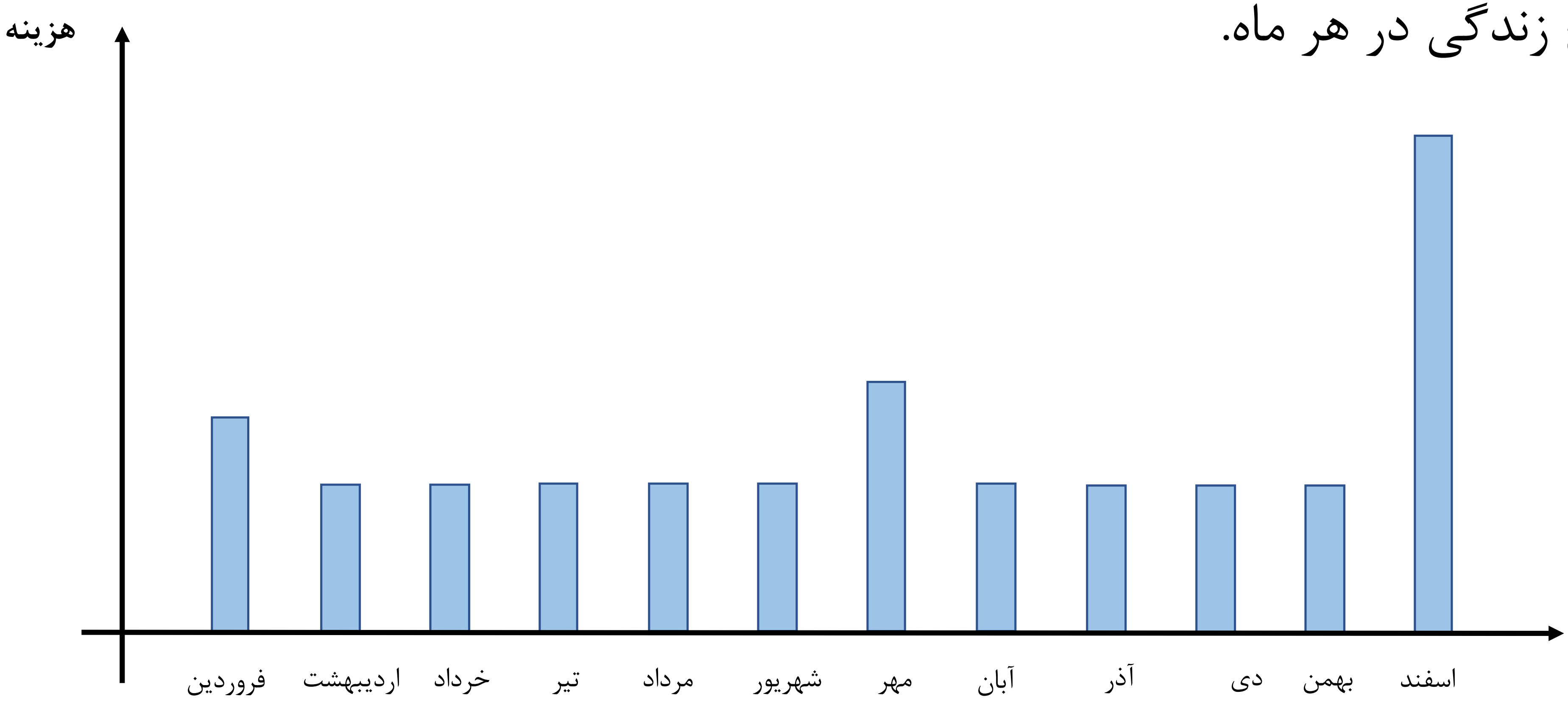
- در تحلیل سرشکنی نشان می‌دهیم که در یک دنباله‌ای از عمل‌ها، هزینه میانگین یک عمل میتواند کم باشد در حالی که انجام آن عمل به صورت تنها ممکن است هزینه بر تلقی شود.

تحلیل سرشکنی (Amortized)

- تفاوت با تحلیل حالت میانگین و بدترین حالت؟

مثال

- مخارج زندگی در هر ماه.



مثال ۱

IUT-ECE

- عملگرهای پشته S:

PUSH(S, x) pushes object x onto stack S.
POP(S) pops the top of stack S and returns the popped object.

- هر کدام با هزینه ۱
- زمان اجرا برای دنباله‌ای از n تا push و pop؟

$$\Theta(n)$$

مثال ۱

- افزودن یک عملگر جدید:

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

مثال ۱

- افزودن یک عملگر جدید:

(a)	(b)	(c)
top → 23		
17		
6		
39		
10	top → 10	
47	47	

MULTIPOP (S,4)   MULTIPOP (S,7)

مثال ۱

- افزودن یک عملگر جدید:

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

- هزینه این عملگر روی پشته‌ای با s عنصر:

$$\min\{s, k\}$$

مثال ۱

- سوال: بیشترین هزینه برای انجام n عمل متوالی (دنباله اعمال) از push، pop و multipop چیست؟ (با فرض شروع از پشته خالی)
  - بیشترین هزینه هر عمل برابر $O(n)$
  - هزینه کل میشود $O(n^2)$

تحلیل بدترین حالت

دقیق؟

مثال ۲

- افزایش یک شمارنده باینری (شروع از صفر):

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

least significant bit

most significant bit

مثال ۲

- افزایش یک شمارنده باینری (شروع از صفر):

```
INCREMENT(A, k)
i = 0
while i < k and A[i] == 1
        A[i] = 0
        i = i + 1
if i < k
        A[i] = 1
```

مثال ۲

- تحلیل در بدترین حالت:
  - بدترین حالت برای increment:

$$\Theta(k)$$

  - پس بعد از n عمل افزایش داریم:

$$O(kn)$$

دقیق؟

تحلیل سرشکنی

- هزینه بدترین حالت در دنباله

تکنیک‌های تحلیل سرشکنی

- سه تکنیک اصلی برای تحلیل سرشکنی:

  - تحلیل انبوهه (aggregate)
  - تحلیل حسابداری (accounting)
  - تحلیل پتانسیل (potential)

تحلیل انبوهه

- در تحلیل انبوهه: در این روش مطابق تعریف جمع هزینه‌های اعمال محاسبه می‌شود و بر تعداد آنها تقسیم می‌شود تا هزینه سرشکن شده به دست آید.

تحلیل انبوهه

- برای مثال ۱ (پشته):
- سوال: بیشترین هزینه برای انجام n عمل متوالی (دنباله اعمال) از push، pop و multipop چیست؟ (با فرض شروع از پشته خالی)

مثال ۱

- عملگرهای پشته S:

PUSH(S, x) pushes object x onto stack S.

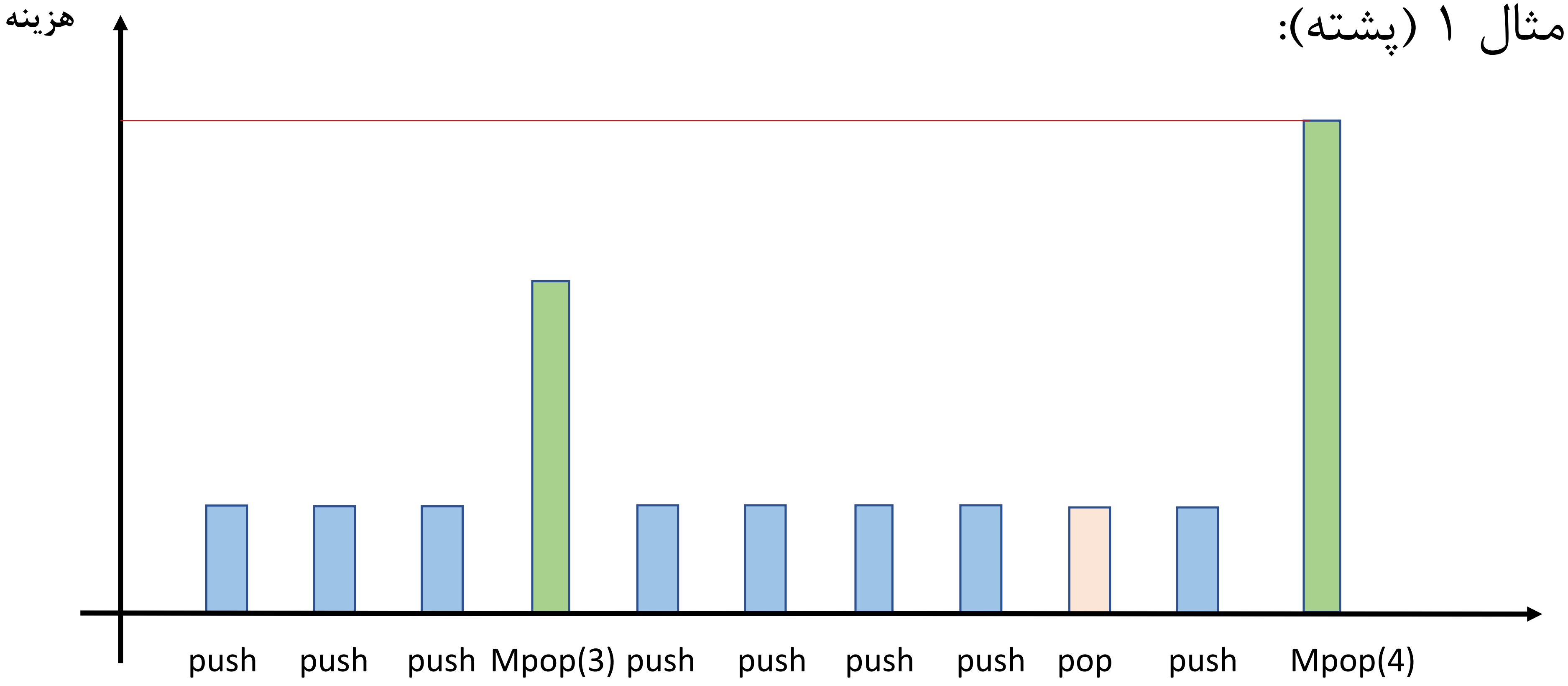
POP(S) pops the top of stack S and returns the popped object.

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

تحلیل انبوهه

IUT-ECE

- برای مثال ۱ (پشته):



تحلیل انبوهه

- برای مثال ۱ (پشته):
- سوال: بیشترین هزینه برای انجام n عمل متوالی (دنباله اعمال) از push، pop و multipop چیست؟ (با فرض شروع از پشته خالی)
  - در نظر گرفتن کل دنباله
  - هر عمل pop یا فراخوانی در multipop تنها زمانی قابل انجام است که از قبل یک push داشته باشیم.
  - حداکثر n بار push داریم.
  - بنابراین هزینه سرشکن برای یک عمل برابرست با $\frac{O(n)}{n} = O(1)$

تحلیل انبوهه

- برای مثال ۲ (شمارنده باینری):
- سوال: بدترین حالت برای increment برای n بار متوالی
  - در نظر گرفتن کل دنباله

مثال ۲

- افزایش یک شمارنده باینری (شروع از صفر):

```
INCREMENT(A, k)
i = 0
while i < k and A[i] == 1
        A[i] = 0
        i = i + 1
if i < k
        A[i] = 1
```

تحلیل انبوهه

IUT-ECE

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

تحلیل انبوهه

IUT-ECE

- برای مثال ۲ (شمارنده باینری):
- سوال: بدترین حالت برای increment برای n بار متوالی
  - در نظر گرفتن کل دنباله
  - زمان اجرای عملگر increment متناسب با تعداد بیتهای تغییریافته است.
  - مثلا A[2] بعد از هر چهار عمل متوالی یکبار عوض میشود.
  - بنابراین هزینه سرشکن برای یک عمل برابرست با

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

تکنیک‌های تحلیل سرشکنی

- سه تکنیک اصلی برای تحلیل سرشکنی:

  - تحلیل انبوهه (aggregate)
  - تحلیل حسابداری (accounting)
  - تحلیل پتانسیل (potential)

تحلیل حسابداری

IUT-ECE

- در تحلیل حسابداری: در این روش به ازای هر عمل مقداری پول پرداخت می‌شود. از این مقدار مبلغی صرف انجام واقعی عمل مورد نظر می‌شود و مازاد به مخزن پول اضافه می‌شود. اگر مقدار پول پرداختی برای انجام آن عمل کافی نباشد از مخزن پول برداشت می‌شود. اگر مخزن هیچ وقت کمبود پول نداشته باشد میزان پول پرداختی برای هر عمل با هزینه سرشکن شده آن عمل متناسب است.

تحلیل حسابداری

Let's denote the actual cost of the ith operation by c_i and the amortized cost of the ith operation by $\widehat{c}_i$. Then you need to have

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

for all sequences of n operations. The total credit stored in the data structure is the difference between the total amortized cost and the total actual cost, or $\sum_{i=1}^{n} \widehat{c}_i - \sum_{i=1}^{n} c_i$.

تحلیل حسابداری

- برای مثال ۱ (پشته):
- سوال: بیشترین هزینه برای انجام n عمل متوالی (دنباله اعمال) از push، pop و multipop چیست؟ (با فرض شروع از پشته خالی)
- فرض برای هزینه سرشکن هر عملگر:

	هزینه واقعی	هزینه سرشکن
PUSH	1 ,	2 ,
POP	1 ,	0 ,
MULTIPOP	$\min\{s,k\}$,	0 .

تحلیل حسابداری

	هزینه واقعی	هزینه سرشکن
PUSH	1 ,	2 ,
POP	1 ,	0 ,
MULTIPOP	$\min\{s, k\}$,	0 .

- میتوان نشان داد:

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

- هزینه کل:

$$O(n)$$

تحلیل حسابداری

IUT-ECE

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 مثال ۲ (شمارنده باینری):

operation	هزینه واقعی	هزینه سرشکن
Bit 0 -> 1	1	2
Bit 1 -> 0	1	0

- چک کردن شرط
- هزینه کل:

$$n \times O(1) = O(n)$$