

نمونه

Amortized
Analysis

تجزیه و تحلیل

bucket sort

مثال دیگر

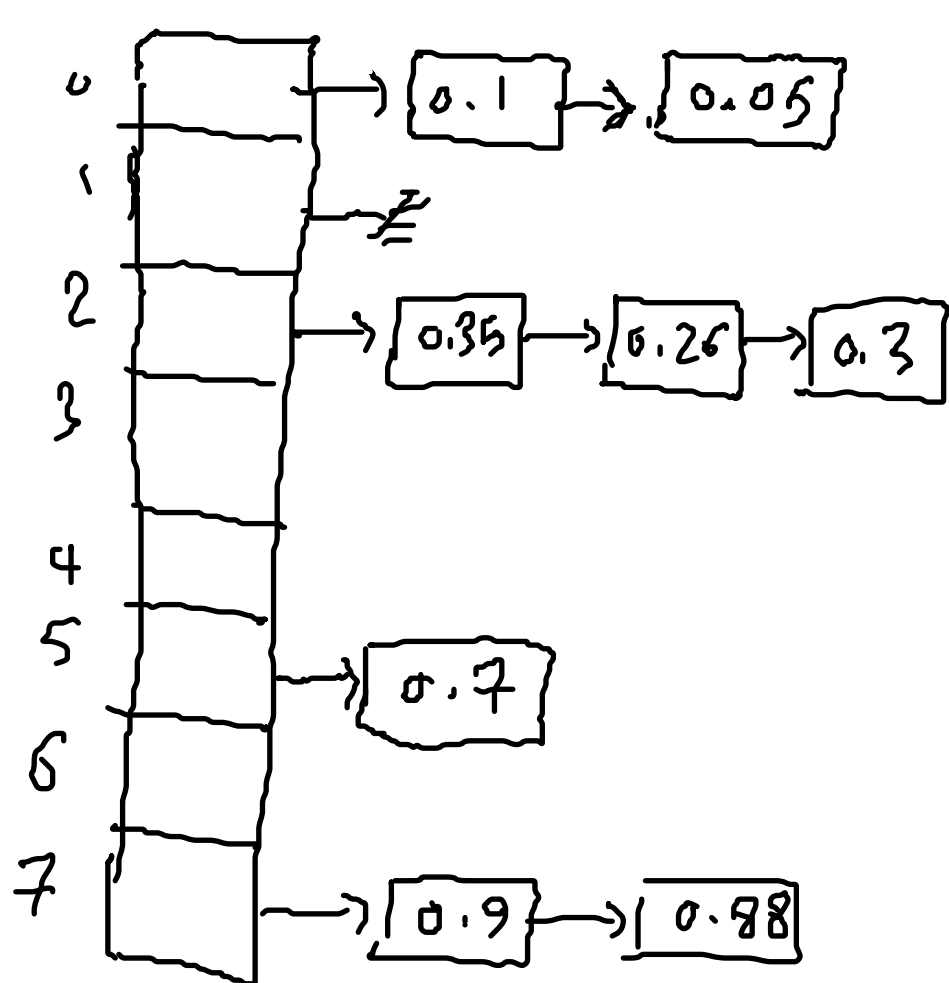
$n=8$
1 2 3 4 5 6 7 8

0.05 , 0.1

0.35 , 0.3 , 0.26

0.7 , 0.9 , 0.88

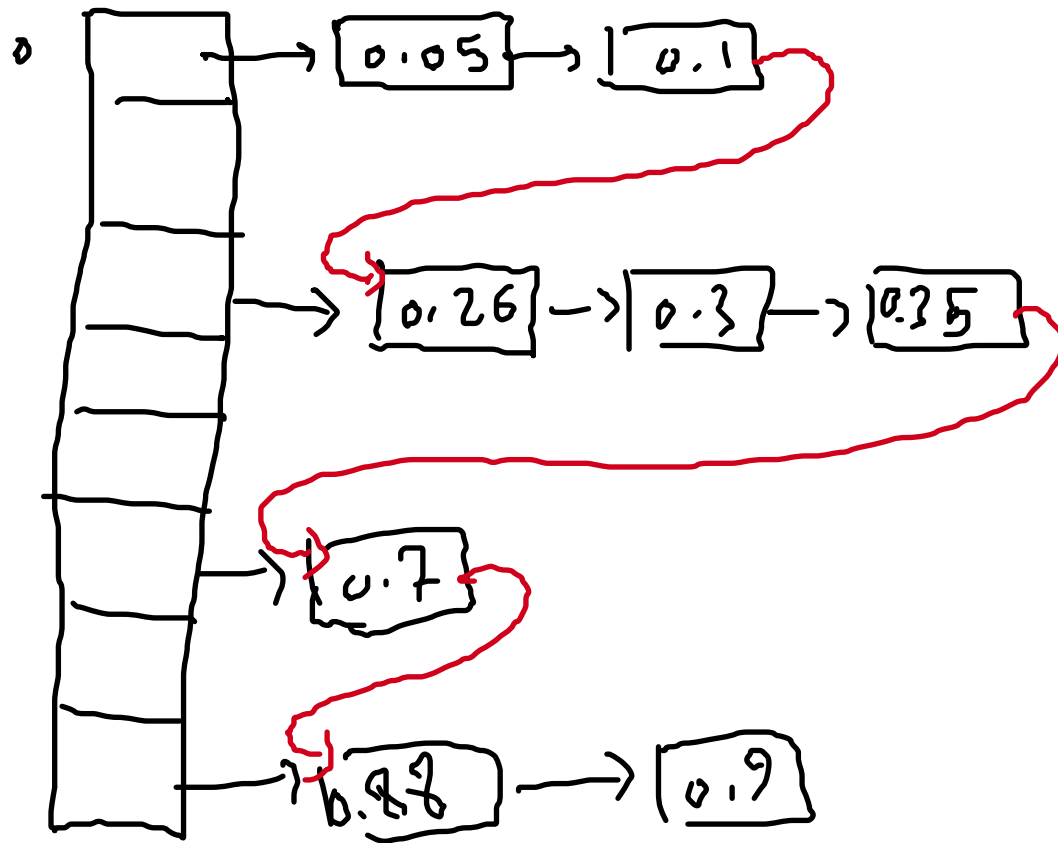
$[n \times A[i]]$ = اندیس بک



1/8
2/8

8/8

: list sort :
↓



تحلیل سرنگن :

اگر n عمل انجام شود و برحسب هزینه زیاد داشته باشد
 و برحسب هزینه کم ، مستقار هزینه زیاد را عملیات
 با هزینه زیاد را ، و بقیه عملیات پخش (سرنگن)
 کرد.

۱۰ عمل

$$\frac{2 \times 6 + 1 \times 8}{10 \text{ عمل}} = 2 \leftarrow \begin{array}{cc} 6 & \text{هزینه} \\ 1 & \text{هزینه} \end{array} \begin{array}{c} ۲ \text{ عمل} \\ ۱ \text{ عمل} \end{array}$$

↑
هزینه سرنگن

هدف تحلیل سرشکن: محاسبه حد بالایی کمتر برای هزینه کار
الگوریتم‌ها است.

مثال) n عمل دارم که مقدار ثابت از این عملیات

در بدترین حالت هزینه $O(n)$ دارند

و هزینه عملیات در بدترین حالت $O(1)$ دارند.
هزینه

هزینه n عمل چقدر است؟

تحلیل کلی و خالی خام: $n \times O(n) = O(n^2)$ هزینه کار n عمل
حد بالایی

$$\text{هزینه سرشکن در عمل} = \frac{O(n^2)}{n} = O(n)$$

C عدد ثابت

باجای سرشکن :

هزینه n عمل

$$C \times O(n) + (n - C) \times O(1) = O(n) + O(n) \\ = O(n)$$

$$\text{هزینه سرشکن} = \frac{\text{هزینه کل}}{\text{تعداد عملیات}} = \frac{O(n)}{n} = O(1)$$

در تحلیل سرشکن سعی بر این است که هزینه کار واقعی را دقیق تر حساب کنیم تا بتوانیم حد بالایی کمتری برای هزینه کار اثبات کنیم.

۳ روش تحلیل شرکت

① روش aggregate analysis = $\frac{\text{شرکت جمع دقیقاً کل هزینه ها}}{\text{تعداد عملیات}} = \text{هزینه یک عمل}$

درین روش هزینه شرکت ها انواع عملیات با هم مساوی هستند.

② روش accounting method

$\text{credit} = \text{هزینه واقعی عمل} - \text{هزینه شرکت عمل}$
یک ساختمان را به دارم که credit هر عمل در آن هزینه مشور

③ روش potential method

مثل روش (۲) است ولی credit عملیات جدا جدا هزینه من شود
بلکه یک تابع پیمایش کل credit نشان

از ۲ میزن براس اراته روشی استفااء دستور:

تکامل aggregate

① میزن stack :

مزینه واقعی $O(1)$ شیء n را در پشته S قرار میده: $push(S, n)$

شیء بالایی پشته را بر مگردان: $pop(S)$: مزینه واقعی $O(1)$

$multi pop(S, k)$

1. while not stack-empty(S) and $k \neq 0$ do

2. $pop(S)$

3. $k \leftarrow k - 1$

مزینه واقعی

در بدنه حالت $multi pop$ مزینه $O(n)$ ← اندازه پشته $\min\{n, k\}$

باعتبار نکی : n عمل داریم که بهر $O(n)$ هستند پس
لزیم n عمل $O(n^2)$ مشور.

باعتبار سرشتن :

* نکته : تا شئی روی میشته $push$ شود میخواند pop شود

تعداد کل عملیات n تا است \leftarrow ~~تعداد کل عملیات~~ $push$
حداکثر n است و هزینه آن $O(n)$

تعداد pop ها کمتر از تعداد $push$ ها است
تعداد $push$ ها است \leftarrow تعداد کل pop ها حداکثر
 n است و هزینه آن $O(n)$

$$\begin{aligned} \text{هزینه کل} &= \text{هزینه } push \text{ ها} + \text{هزینه } pop \text{ ها} \\ &= O(n) + O(n) \\ &= O(n) \end{aligned}$$

$$\frac{\text{هزینه سرگشتن}}{\text{تعداد}} = \frac{\sigma(n)}{n} = O(1)$$

↓

هزینه سرگشتن push , pop , multipop
 $O(1)$ است.

مثال 2) binary counter

n عمل Increment

عمل Increment

هر سینه تغییر بیت	ک بیت
	0 0 0 0
1	0 0 0 1
2	0 0 1 0
1	0 0 1 1
3	0 1 0 0
1	0 1 0 1
2	0 1 1 0
1	0 1 1 1
4	1 0 0 0
⋮	⋮

در هر حرف از عملیات INCY
بیتها زیاد تغییر میکند
 $O(k)$ بیت تغییر میکند

تکلیف کلی : n عمل داریم
که هر حرف $O(k)$ هزینه دارد
پس هزینه n عمل دستور
 $O(n.k)$

تحلیل سرشکن :

بیت اول در هر عمل Increment تغییر میکند

بیت دوم یک در میان بیت سوم 4 تا در میان والی آخر

$$\text{میزان تغییر بیتها برای } n \text{ عمل} = \sum_{i=0}^{K-1} \frac{n}{2^i} \leq n \sum_{i=0}^{\infty} \frac{1}{2^i} = 2n$$

$$\text{میزان سرشکن} = \frac{\text{میزان کل}}{n} \leq \frac{2n}{n} = 2$$

میزان سرشکن در عمل increment $\approx (1) 0$ است.

روشی accounting :

اول برابر هر عمل هزینه سرشکن تعریف میکنیم

ند C_i هزینه واقعی عمل نام \hat{C}_i هزینه سرشکن عمل نام

$$\sum_{i=1}^n \hat{C}_i \approx \sum_{i=1}^n C_i$$

آنچه مجموع هزینه‌های سرشکن است حد بالا برابر

مجموع هزینه‌های واقعی است.

مثال ۱) عملیات stack
هزینه سرشتن؟

push

هزینه واقعی

1

2

pop

1

0

multipop

$\min(k, s)$

0

ایمه :

هزینه pop را قبلاً موقع push پرداخت کنیم

$$\text{credit} = \sum \hat{c}_i - \left\{ \begin{array}{l} c_i \\ \text{stack} \end{array} \right\} = \text{تعداد عناصر داخل}$$

بنابراین هزینه کار سرشتن حد بالایی هزینه کار واقعی است.

2n < مجموع هزینه‌های
ساختن مجموع هزینه‌های
واقعی