AMORTİZE ANALİZ

Başlangıçta D(0) gibi bir veri vardır ve bu veri boştur. Birde işlemler kümesi vardır.
D(1) verisine ulaşmak için bir işlem seçilir.

$D(0) \rightarrow D(1) \rightarrow D(2) \rightarrow \ldots\ldots\ldots \rightarrow D(n)$

T(n): Toplam en kötü durum zamanı olsun ve buradan işlem başına zaman T(n)/n olur.

Tümleşik Metod

Her n için, n tane işlemin en kötü durum zamanı T(n) olur. Ortalama maliyet veya amortize maliyet T(n)/n olur. Birden fazla işlem olsa dahi amortize maliyet her işlem başına olan maliyettir.

Örnek: Yığıt

İlk örnek yığıt veri yapısına yeni bir işlem eklenmesi ile ele alınsın.

$$PUSH(S,x) \rightarrow \Theta(1)$$
$$POP(S) \rightarrow \Theta(1)$$

Her birinin maliyeti 1 olduğu kabul edilsin. n tane PUSH ve POP işleminin maliyeti n olur ve
$T(n) = \Theta(n)$ olur.

$$MULTIPOP(S,k) = \left\{ \begin{array}{l} \text{Eğer S'de k'dan fazla eleman varsa, k tanesini çıkarır.} \\ \text{Eğer k'dan az eleman varsa, hepsini çıkarır} \end{array} \right\}$$

Algoritma 16.1=MultiPOP (S,k)

▷Yerel Değişkenler: $k \in Z$ ve S yığıtı.

1. S yığıtı boş değil ve $k \neq 0$ olduğu sürece
   a. POP(S)
   b. $k \leftarrow k-1$

Eğer S' de s tane eleman varsa, MULTIPOP(S,k) algoritmasında döngünün çalışmasında min(S,k) tane iterasyon vardır. Her iterasyonda bir sefer POP algoritması icra edilmektedir. Böylece bu algoritmanın maliyeti min(S,k) olur ve zaman bağıntısı POP fonksiyonunun maliyetinin lineer bağıntısı olur.

Başlangıçta boş olan bir S için karışık olarak PUSH, POP ve MULTIPOP işlemleri uygulansın. MULTIPOP işleminin en kötü durum maliyeti O(n) olur, çünkü S en fazla n tane eleman içerir. Bundan dolayı yığıtın işlemlerinden herhangi birinin en kötü durum zamanı O(n) olur ve böylece n tane durum zamanı $O(n^2)$ olur. Çünkü O(n) tane MULTIPOP işlemi olabilir ve her birinin maliyeti O(n) olur. Analiz doğru olmasına rağmen bir işlem için verilen sınır çok geniş bir aralık tanımlamaktadır.

Tümleşik metot kullanılarak daha iyi bir sınır değeri elde edilebilir. n tane PUSH, POP ve MULTIPOP işlem dizisinin maliyeti en fazla O(n) olur. Çünkü her eleman bir kez PUSH işlemine tabi tutulur ve bir kez de POP işlemine tabi tutulur. Böylece POP işleminin icra edilme sayısı (MULTIPOP içerisinde çağrılmalarda dahil) en fazla PUSH işlem sayısına eşit olur. Böylece n tane PUSH, POP ve MULTIPOP işlemler dizisinin maliyeti

$$T(n)=O(n)$$

olur. Böylece bir işlemin maliyeti

$$\frac{T(n)}{n} = O(1)$$

olur.

Örnek: İkili sayıcıyı artırma

Sıfırdan başlayan ve k-bit olan bir sayıcının değerinin artırılması problemi düşünülsün

A[0…. k-1] : bitler dizisi ve Uzunluk(A)=k olur. Sayıcıda saklanan değer x olsun ve en önemsiz biti A[0] olur ve en önemli biti A[k-1]'de olur. Bu sayıcıda herhangi bir andaki sayısal değer

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

olur. Başlangıçta x=0 olur ve böylece j=0,1,2,3……. k-1 için A[j]=0 olur. Bu sayıya 1(mod 2^k) sayısının sayıcıya eklenmesi için aşağıdaki algoritma kullanılır.

Algoritma 16.2=Artır (A)

▷Yerel Değişkenler: A dizisi.

1. j←0
2. j<uzunluk(A) ve A[j]=1 olduğu sürece
3.   A[j] ←0
4.   j← j+1
5. eğer j<uzunluk(A) ise
6.   A[j] ←1

Bu algoritmanın analizi yapılacak olursa, en kötü durumda bir sefer artırmanın maliyeti Θ(k) olur. En kötü durumda A dizisinin bütün elemanları 1 olduğunda ortaya çıkar. Böylece n tane artırma işleminin maliyeti $O(n^k)$ olur. Bu sınır değeri geniş bir aralık içerir. Daha dar bir aralık belirlenebilir

İlk adım $A[0] \leftarrow 1$ olur

İkinci adım

$A[0] \leftarrow 0$
$A[1] \leftarrow 1$

Üçüncü adım

$A[0] \leftarrow 1$

Dördüncü adım

$A[0] \leftarrow 0$
$A[1] \leftarrow 0$
$A[2] \leftarrow 1$

A[0] her defasında değer değiştirir.n sefer artırma işleri yapılırsa A[1] elemanı $\left\lfloor \frac{n}{2} \right\rfloor$ sefer değer değiştirir. A[2] elemanı $\left\lfloor \frac{n}{4} \right\rfloor$ defa değer değiştirir.

Eğer $j > \lfloor \lg n \rfloor$ ise A[j] hiçbir zaman değer değiştirmez. Toplam değer değişimi

$$\sum_{j=0}^{\lfloor \lg n \rfloor} \left\lfloor \frac{n}{2^j} \right\rfloor < n \sum_{j=0}^{\infty} \frac{1}{2^j} = 2n$$

Böylece toplam n adım zamanın en kötü durum zamanı O(n) olur ve bir adım zamanı

$$\frac{O(n)}{n} = O(1)$$

olur.

16.2. Muhasebe Yöntemi

Her işleme bir maliyet yüklenir. Farklı işlemlere, farklı maliyetler yüklenir. Buna o işlemin amortize maliyeti denir. Amortize maliyeti, gerçek maliyetin geçtiği zaman, aradaki fark kredi denilen bir veri yapısına yüklenir. Bu değer daha sonraki işlemlerde amortize maliyeti geçek maliyetten küçük olan işlemlere eklenir. Böylece amortize maliyeti gerçek maliyet ve kredi arasında paylaştırılır(ya depolanır yada tüketilir). Tümleşik yöntemde bütün işlemlerin maliyeti aynı iken, burada farklıdır.

Amortize maliyet çok dikkatli seçilmelidir.
En kötü durumda her işlemin maliyetinin küçük(ortalama maliyet) olduğu gösterilirse, toplam amortize maliyet bir üst sınır belirlemelidir. Kredilerin negatif olmamasına dikkat edilmelidir.

Yığıt işlemler

Yığıtın boyutu s olmak üzere

PUSH	1
POP	1
MULTIPOP	min(k,s)

olur. Amortize maliyetler

PUSH	2
POP	0
MULTIPOP	0 ← sabit;

MULTIPOP işleminin gerçek maliyet değişkendir. Bütün işlemlerin maliyeti O(1) olur ve genelde amortize maliyet asimptotik olarak değişebilir.

Her PUSH işleminde 1 birim maliyet gerçek maliyet için harcanırken, 1 birim maliyet kredi olarak o elemanda saklanır. Böylece n tane PUSH, POP ve MULTIPOP işleminin maliyeti O(n) olur.

İkili sayıcı artırma

Sayıcının ilk değeri sıfır olmak üzere bu problemde icra zamanı değer değiştiren bit sayısı ile orantılıdır. Her bit değişiminin ($0 \rightarrow 1$) maliyeti 2 olsun. Her bitin $0 \rightarrow 1$ değişiminde 1 birim maliyet gerçek maliyet için harcanırken, 1 birim maliyet kredi olarak saklanır ve bu değer $1 \rightarrow 0$ işleminin maliyetini karşılar. n tane artışın maliyeti O(n) olur.

16.3. Potansiyel Yöntemi

Belli bir değerde kredi tutmak yerine potansiyel enerji veya potansiyel olduğu kabul edilir. Bu enerji belli bir veri yapısına değil, bütün veriye aittir. İlk olarak D_0 gibi bir veri ile başlanır ve n tane işlem uygulanacaktır. i=1,2,3,…………,n için i. işlemin gerçek maliyeti C_i olsun ve D_i' de i tane işlemin D_0 uygulanmasından elde edilen veri değeri olsun ve

$$D_i \xleftarrow{i} D_{i-1} \xleftarrow{i-1} D_{i-2} \xleftarrow{i-2} \cdots \xleftarrow{2} D_1 \xleftarrow{1} D_0$$

şeklinde gösterilsin. Φ : Potansiyel fonksiyonu ve $\Phi: D_i \rightarrow \Phi(D_i) \in R$ şeklinde tanımlansın ve bu D_i veri yapısının potansiyelini verir. Çünkü potansiyel fonksiyonu verinin değerini veya durumunu parametre olarak alır ve sonuç olarak bir gerçel sayı verir ve bu sayı değeri verinin potansiyelini gösterir. i işlemi için amortize maliyet $\hat{C}_i$ ile gösterilir ve

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

şeklinde tanımlanır. Amortize maliyet, gerçek maliyet ile veri yapısındaki potansiyel artışının toplamına eşittir. n tane işlemin amortize maliyeti

$$\sum_{i=1}^{n} \hat{C}_i = \underbrace{\sum_{i=1}^{n} \left(C_i + \Phi(D_i) - \Phi(D_{i-1})\right)}_{\text{teleskop serisi}}$$

şeklinde tanımlanır ve

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

olur. Eğer Φ potansiyel fonksiyonu $\Phi(D_n) \geq \Phi(D_0)$ şeklinde tanımlanabilirse,

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

üst sınırı teşkil eder. Pratikte her zaman ne kadar işlem gerçekleştirileceği bilinmediğinden bütün i'ler için

$\Phi(D_i) \geq \Phi(D_0)$

olmalıdır. Genellikle $\Phi(D_0)=0$ olarak tanımlanır ve

$\forall i$, $\Phi(D_i) \geq 0$

olur. Eğer

$\Phi(D_i) - \Phi(D_{i-1}) > 0$

ise, Ĉi' de aşırı artış var demektir ve verinin potansiyeli artar. Eğer

$\Phi(D_i) - \Phi(D_{i-1}) < 0$

ise, Ĉi'de aşırı azalma var demektir ve verinin potansiyeli her işlemden(i. işlem) sonra azalır.

Yığıt işlemleri

Φ: yığıtta bulunan eleman sayısını versin. D_0 için $\Phi(D_0)=0$ olur, çünkü yığıt boştur. Böylece $\forall i$ için $\Phi(Di) \geq 0$ olur. n tane işlemin amortize maliyeti bir üst sınır teşkil eder. Eğer i. işlem PUSH ise (s tane eleman var)

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

olur. Amortize maliyet

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

olur. Eğer i. işlem MULTIPOP(S,k) ise, k'=min(k,S) tane eleman S' ten çıkarılır. Gerçek maliyet k' olur ve potansiyel fark ise $\Phi(D_i)-\Phi(D_{i-1})=-k'$ olur. Böylece MULTIPOP işleminin amortize maliyeti

$$\hat{C}_i=C_i+\Phi(D_i)-\Phi(D_{i-1})$$
$$= k'-k'$$
$$=0$$

Benzer bir şekilde POP işleminin amortize maliyeti 0 olur. Her üç işlemin amortize maliyeti O(1) olduğundan n tane işlem için toplam amortize maliyet O(n) olur. $\Phi(D_i)\geq\Phi(D_0)$ olduğundan toplam amortize maliyet üst sınır teşkil eder. Böylece n tane işlemin en kötü durum maliyeti O(n) olur.

İkili sayıcı artırma

Sayıcı için potansiyelin tanımlanması gerekir. i. artırma işleminin potansiyeli b_i olsun ve b_i sayıcıdaki 1 sayılarının toplamıdır. Artırma işleminin amortize maliyeti hesaplansın. i. işlemde t_i tane $1\rightarrow 0$ olsun. Bu işlemin gerçek maliyeti en fazla t_{i+1} olur. i. işlemden sonra sayıcıdaki 1 sayısı

$$b_i\leq b_{i-1} - t_i+1$$

olur ve potansiyel farkı

$$\Phi(D_i)-\Phi(D_{i-1}) \leq (b_{i-1} - t_i + 1) - b_{i-1}$$
$$= 1 - t_i$$

olur ve amortize maliyet

$$\hat{C}_i = C_i + \Phi(D_i) - \Phi(D_{i-1})$$
$$\leq (t_i+1)+(1-t_i)$$
$$=2$$

olur.

Sayıcı sıfırdan başlarsa $\Phi(D_0)=0$ olur. $\forall i$ için $\Phi(D_i)\geq 0$ olduğundan n tane artırma işleminin amortize maliyet toplamı, toplam gerçek maliyet için bir üst sınır teşkil eder. Böylece n tane işlemin en kötü durum üst sınırı O(n) olur.

Eğer sayıcı sıfırdan başlamazsa, b_0 tane 1 vardır. n tane artırma işleminden sonra b_n tane 1 olur ve $0\leq b_0$, $b_n \leq k$ olur.

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

$\hat{C}_i \leq 2$ olduğundan ve $\Phi(D_0)=b_0$ ve $\Phi(D_n)= b_n$ gerçek maliyet (n tane işlemden sonra)

$$\sum_{i=1}^{n} C_i \leq \sum_{i=1}^{n} 2 + b_n - b_0$$
$$= 2n - b_n + b_0$$

olur. $b_0 \leq k$ olduğundan eğer $n=\Omega(k)$ işlem uygulanırsa, toplam gerçek maliyet O(n) olur.

Dinamik tablolar

Bazı uygulamalarda tabloda kaç tane nesne olduğu bilinmez. Zaman içerisinde tablonun boyutu az gelirse, boyutu daha büyük olan bir tablo oluşturulur ve yeni değerler ile eski tablodaki değerler yeni tabloya kopyalanır. Benzer şekilde tablodan değer silindiğinde, tablonun boyutu küçülebiliyorsa, tablonun boyutu yarıya indirilir. Amaç, bu tablolara eleman ekleme ve eleman silme işlemlerinin amortize maliyetlerinin O(1) olduğunun gösterilmesidir. Gerçek maliyetlere gelince tablonun genişlemesi veya daralması işlemlerinin maliyeti O(1)'den daha büyüktür.

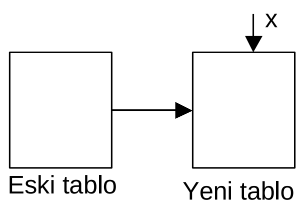
İşlemler

T.EKLE→bir eleman ekler

T.SİL→bir eleman siler ve bir slot serbest bırakır.

Çırpı tablosunun kullanıldığını kabul edelim ve $\alpha(T)$: yükleme faktörü olsun. Boş tablonun boyutu 0 ve $\alpha(T)=1$ olarak kabul edilir. Eğer bir dinamik tablonun $\alpha(T)$ alttan sınırlı ise, kullanılmayan slot sayısı bütün boyutun bir sabit oranıdır.

Eğer bütün slotlar kullanılırsa,bu tabloya dolu tablo denir veya $\alpha(T)=1$ olur. Dolu bir tabloya eleman eklenmek istenirse, bu işlem kesilir ve iptal edilir. Başka bir yöntemde ise tablo genişletilir.



Genelde boyut iki katına çıkarılır. Eğer sürekli T.EKLE işlemi uygulanırsa, $\alpha(T)\geq 1/2$ olur. Tabloya eleman ekleme algoritması Algoritma X.X' te görülmektedir.

Her ekleme işleminin maliyeti 1 olur. T.EKLE algoritmasının zamanı bir elemanı ekleme zamanının lineer bağıntısı olur. Boş tablodan başlayan T.EKLE işlem dizisinin analizi yapılabilir. i.işlem için C_i=? Eğer tabloda boş yer varsa, $C_i=1$ olur. Eğer tablo dolu ve genişletme işlemi uygulanırsa, $C_i=i$ olur. Bir tane eleman ekleme yapılır ve i-1 tane eleman eski tablodan kopyalanır ve toplamda eklenen eleman sayısı i olur. Eğer n tane işlem uygulanırsa, en kötü durumda bir işlemin maliyeti O(n) olur ve toplam maliyet $O(n^2)$ olur. Bu sınır sıkı bir sınır değildir. Eğer i-1=2^k gibi bir değer ise, i. işlemde tablo genişler. Amortize maliyet O(1) olur.

Algoritma 16.3=T.EKLE(T,x)

▷ T:tablo
no(T): tablodaki eleman sayısı
boyut(T):tablonun içermiş olduğu slot sayısı
Tablo boş iken no(T) = boyut(T)=0 olur.

```
eğer boyut(T)= 0 ise
▷    T 'ye bir slot ekle
     boyut(T)←1
eğer no(T)=boyut(T) ise
     yeni T' oluştur ve
boyut(T')←2.boyut(T)
     T→T' bütün kayıtları taşı
     T tablosunu serbest bırak
     T←T'
     boyut(T) ← 2.boyut(T)
T'ye x'i ekle
no(T) ←no(T)+1
```

<u>Tümleşik yönteme göre:</u>

$$C_i = \begin{cases} i & i-1=2^k \text{ ise} \\ 1 & \text{aksi halde} \end{cases}$$

n tane T.EKLE işleminin toplam maliyeti

$$\begin{aligned} \sum_{i=1}^{n} C_i &\le n + \sum_{j=0}^{\lfloor \lg n \rfloor} 2^j \\ &< n + 2n \\ &= 3n \end{aligned}$$

olur. Toplam amortize maliyet 3n olduğundan bir işlemin amortize analizi sonucunda amortize maliyetinin 3 olduğu rahatlıkla görülebilir.

Muhasebe yöntemi

Ekleme maliyeti 3 olsun (ekleme,taşıma). Toplam boyutu m olsun (genişletmeden hemen sonra), bu durumda, tabloda m/2 tane eleman vardır ve kredi yoktur. Bir eleman eklenir, onun için maliyeti 1 olur.

1:kredi
1:daha önce eklenmiş elemanlardan birine kredi olarak ver.

Tablonun dolması için m/2 tane eklemeye gerek vardır, böylece tablo m tane eleman içerir ve dolu olur. Her elemanın tabloda genişleme olursa, taşırma maliyeti vardır.

Potansiyel yöntemi

n tane T.EKLE işlemi analiz edilebilir. Eleman silme işlemi O(1) olacak şekilde tasarlanabilir. Bunun için Φ tanımlanmalıdır ve her tablo genişlemesinde Φ sıfır olmalıdır. Zamanla tablo dolu olduğu zaman tablonun boyutuna eşit olur.

$\Phi(T)=2.no(T)-boyut(T)$

şeklinde tanımlansın. Genişlemeden hemen sonra no(T)=boyut(T)/2 olduğundan $\Phi(T)=0$ olur.

Genişlemeden önce no(T)=boyut(T) olduğundan $\Phi(T)$=boyut(T) olur. Başlangıçta tablo boş olduğundan ilk değer sıfır olur ve her zaman için no(T)≥boyut(T)/2 olduğundan $\Phi(T) \geq 0$ olur. Böylece n tane T.EKLE işleminin amortize maliyeti üst sınır teşkil eder. i. T.EKLE işlemi analiz edilebilir. Analiz için no(T)=n(T) ve boyut(T)=b(T) olsun. i. işlemden sonra tablodaki eleman sayısı $n_i(T)$ olsun ve $b_i(T)$'de boyutu olsun ve Φ_i i.işlemden sonra potansiyeli göstersin.

Başlangıçta $n_0(T)=0$ ve $b_0(T)=0$ ve $\Phi_0 = 0$ olur. Eğer i.işlem tablonun genişlemesine neden olmuyorsa,

$$\begin{aligned}\hat{C}_i &= C_i + \Phi_i - \Phi_{i-1}\\ &= 1 + \left(2n_i(T) - b_i(T)\right) - \left(2n_{i-1}(T) - b_{i-1}(T)\right)\\ &= 1 + \left(2n_i(T) - b_i(T)\right) - \left(2\left(n_i(T) - 1\right) - b_i(T)\right)\\ &= 3\end{aligned}$$

olur. Eğer i. işlem tablonun genişlemesine sebep oluyorsa,

$$\frac{b_i(T)}{2} = b_{i-1}(T) = n_i(T) - 1$$

olur. İşlemin amortize maliyeti

$$\begin{aligned}\hat{C}_i &= C_i + \Phi_i - \Phi_{i-1}\\ &= n_i(T) + \left(2n_i(T) - b_i(T)\right) - \left(2n_{i-1}(T) - b_{i-1}(T)\right)\\ &= n_i(T) + \left(2n_i(T) - \left(2n_i(T) - 2\right)\right) - \left(2\left(n_i(T) - 1\right) - \left(n_i(T) - 1\right)\right)\\ &= n_i(T) + 2 - \left(n_i(T) - 1\right)\\ &= 3\end{aligned}$$

olur. Tablodan eleman silme için eğer $n_i(T)>b_i(T)/2$ ise, sadece eleman silinir. Eğer $n_i(T)=b_i(T)/2$ ise, silme işlemi tablonun daralmasına sebep olur. İki tane özelliğin korunmasına dikkat edilir.

- α(T)'nin belli bir değerin altına düşmesine izin verilmez(alttan sabit ile sınırlı)
- Amortize maliyet üstten bir sabit sınırlıdır.

Maliyet temel ekleme ve silme işlemleri cinsinden ifade edilebilir (bir ölçüsü olarak). $n=2^k$ olmak üzere n tane işlem gerçekleştiriliyorsa, n/2 tane ekleme olsun ve toplam maliyet Θ(n) olur. Ekleme işlemlerinin sonunda n(T)=b(T)=n/2 olur. İkinci n/2 işlemler için aşağıdaki dizi uygulansın.

I,D,D,I,I,D,D,I,I…………,

İlk işlem tablonun boyutunun n çıkmasına sebep olur. Ondan sonraki iki silme işlemi tablonun tekrar n/2 boyutuna daralmasına sebep olur.

I,I→ b(T)=n
D,D→b(T)=n/2
…….

Her genişletme ve daraltma işleminin maliyeti $\Theta(n)$ olur ve $\Theta(n)$ işlem vardır. Böylece n tane işlemin maliyeti $\Theta(n^2)$ olur ve bir işlemin amortize maliyeti $\Theta(n)$ olur.

Bu problemden kaçınmak için $\alpha(T)<1/2$ olmasına izin verilir. Dolu tabloya eleman eklenmesi istenirse, genişleme olur, fakat örneğin $\alpha(T)<1/4$ olduğunda daralma olur. Böylece yükleme faktörü alttan bir sabit ile sınırlıdır. n tane T.EKLE ve T.SİL işlemin maliyet analizi yapılsın. Her genişlemeden sonra ve daralmadan sonra $\Phi=0$ olarak tanımlanır.

$$\left.\begin{aligned} \alpha(T) &= \frac{1}{4} \\ \alpha(t) &= 1 \end{aligned}\right\} \quad \text{ise } \Phi = 0 \text{ olur.}$$

Diğer durumda $\alpha(T)=n(T)/b(T)$ ve boş tablo için $n(T)=b(T)=0$ ve $\alpha(T)=1$ olur. Böylece her zaman

$n(T)= \alpha(T)*b(T)$

olur. Tablo boş olsun veya olmasın potansiyel fonksiyonu

$$\Phi(T) = \begin{cases} 2n(T) - b(T) & \alpha(T) \geq 1/2 \\ \dfrac{b(T)}{2} - n(T) & \alpha(T) < 1/2 \end{cases}$$

Boş tablonun potansiyeli sıfır olur. Potansiyel hiçbir zaman negatif olmaz. Böylece toplam amortize maliyet (işlemler dizisi için) gerçek maliyet üzerinde Φ göre üst sınır teşkil eder. Potansiyel fonksiyonun bazı özelliklerine bakılabilir.

- $\alpha(T)=1/2 \rightarrow \Phi(T)=0$
- $\alpha(T)=1 \rightarrow n(T)=b(T) \rightarrow \Phi(T)=n(T)$
- $\alpha(T)=1/4 \rightarrow b(T)=4n(T) \rightarrow \Phi(T)=n(T)$

n tane T.EKLE ve T.SİL işlemi analiz edilsin ve C_i gerçek maliyeti gösterin. $\hat{C}_i$ ise Φ göre amortize maliyet olsun.

$n_i(T)$:i. işlemden sonra n(T)
$b_i(T)$:i. işlemden sonra b(T)
$\alpha_i(T)$:i. işlemden sonra α(T)

$$\Phi_i = \begin{cases} 2n_i(T) - b_i(T) & \alpha_i(T) \geq \frac{1}{2} \\ \frac{b_i(T)}{2} - n_i(T) & \alpha_i(T) < \frac{1}{2} \end{cases}$$

Başlangıçta $n_0(T)=0$, $b_0(T)=0$ ve $\alpha_0(T)=1$ ise, $\Phi_0= 0$ olur.

i.işlem T_EKLE olsun. Eğer $\alpha_{i-1}(T)\geq 1/2$ ise, daha önce yapılan analizin aynısı yapılır. Tablo genişlemiş veya genişlememiş önemli değil, $\hat{C}_i$ en fazla 3 olur. Eğer $\alpha_{i-1}(T)<1/2$ ise, tablo genişleyemez ve amortize maliyet

$$\begin{aligned} \hat{C}_i &= C_i + \Phi_i - \Phi_{i-1} \\ &= 1 + \frac{b_i(T)}{2} - n_i(T) - \frac{b_{i-1}(T)}{2} + n_{i-1}(T) \\ &= 1 + \frac{b_i(T)}{2} - n_i(T) - \frac{b_i(T)}{2} + n_i(T) \\ &= 0 \end{aligned}$$

olur. Eğer $\alpha_{i-1}(T)<1/2$ ve $\alpha_i(T)\geq 1/2$ ise,

$$
\begin{aligned}
\hat{C}_i &= C_i + \Phi_i - \Phi_{i-1} \\
&= 1 + \left(2n_i(T) - b_i(T)\right) - \left(\frac{b_{i-1}(T)}{2} - n_{i-1}(T)\right) \\
&= 1 + \left(2\left(n_{i-1}(T) + 1\right) - b_{i-1}(T)\right) - \left(\frac{b_{i-1}(T)}{2} - n_{i-1}(T)\right) \\
&= 3n_{i-1}(T) - \frac{3}{2} b_{i-1}(T) + 3 \\
&= 3\alpha_{i-1}(T) b_{i-1}(T) - \frac{3}{2} b_{i-1}(T) + 3 \\
&< \frac{3}{2} b_{i-1}(T) - \frac{3}{2} b_{i-1}(T) + 3 \\
&= 3
\end{aligned}
$$

Böylece T.EKLE işleminin amortize maliyeti en fazla 3 olur. Eğer i.işlem T.SİL ise

$n_i(T)=n_i(T)-1$

olur. Eğer $\alpha_{i-1}(T)<1/2$ ise, daralma olup olmayacağının göz önüne alınması gerekir. Eğer daralma yoksa $b_i(T)=b_{i-1}(T)$ olur ve amortize maliyet

$$
\begin{aligned}
\hat{C}_i &= C_i + \Phi_i - \Phi_{i-1} \\
&= 1 + \frac{b_i(T)}{2} - n_i(T) - \frac{b_{i-1}(T)}{2} + n_{i-1}(T) \\
&= 1 + \frac{b_i(T)}{2} - n_i(T) - \frac{b_i(T)}{2} + n_i(T) + 1 \\
&= 2
\end{aligned}
$$

olur. Eğer $\alpha_{i-1}(T)<1/2$ ve i.işlem tablonun daralmasına sebep oluyorsa, $C_i=n_i(T)+1$ olur, çünkü $n_i(T)$ tane kayıt taşındı ve 1 tanesi silindi. Buradan,

$$\frac{b_i(T)}{2} = \frac{b_{i-1}(T)}{4} = n_i(T) + 1$$

olur.

$$\begin{aligned}
\hat{C}_i &= C_i + \Phi_i - \Phi_{i-1} \\
&= \left(n_i(T) + 1\right) + \left(\frac{b_i(t)}{2} - n_i(T)\right) - \left(\frac{b_{i-1}(T)}{2} - n_{i-1}(T)\right) \\
&= \left(n_i(T) + 1\right)\left(\left(n_i(T) + 1\right) - n_i(T)\right) - \left(\left(2n_i(T) + 2\right) - \left(n_i(T) + 1\right)\right) \\
&= 1
\end{aligned}$$

olur.