

racji, przez $size_i$ – rozmiar tablicy po i -tej operacji, a przez Φ_i – odpowiedni potencjał tablicy. Początkowo mamy $num_0 = 0$, $size_0 = 0$ oraz $\Phi_0 = 0$.

Jeśli i -ta operacja TABLE-INSERT nie wymusza powiększenia tablicy, to $size_i = size_{i-1}$, a koszt zamortyzowany operacji wynosi

$$\begin{aligned}\widehat{c}_i &= c_i + \Phi_i - \Phi_{i-1} \\ &= 1 + (2 \cdot num_i - size_i) - (2 \cdot num_{i-1} - size_{i-1}) \\ &= 1 + (2 \cdot num_i - size_i) - (2(num_i - 1) - size_i) \\ &= 3.\end{aligned}$$

Jeżeli natomiast i -ta operacja powoduje powiększenie tablicy, to wówczas mamy $size_i = 2 \cdot size_{i-1}$ i $size_{i-1} = num_{i-1} = num_i - 1$, skąd wynika, że $size_i = 2 \cdot (num_i - 1)$. Tak więc koszt zamortyzowany jest równy

$$\begin{aligned}\widehat{c}_i &= c_i + \Phi_i - \Phi_{i-1} \\ &= num_i + (2 \cdot num_i - size_i) - (2 \cdot num_{i-1} - size_{i-1}) \\ &= num_i + (2 \cdot num_i - 2 \cdot (num_i - 1)) - (2(num_i - 1) - (num_i - 1)) \\ &= num_i + 2 - (num_i - 1) \\ &= 3.\end{aligned}$$

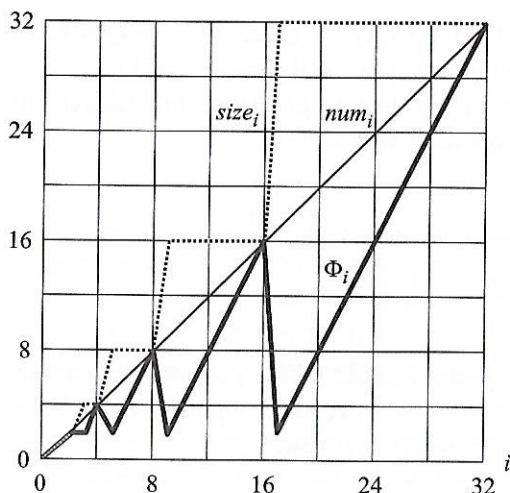
Na rysunku 17.3 są pokazane wartości num_i , $size_i$ oraz Φ_i jako funkcje i . Zwróć uwagę, jak jest „zużywany” potencjał na zapłacenie za operacje powiększenia tablicy.

17.4.2. Powiększanie i zmniejszanie tablicy

W ramach implementacji operacji TABLE-DELETE wystarczy usuwać podany element z tablicy. Jeśli jednak współczynnik zapełnienia tablicy bardzo się zmniejszy, dobrze będzie **zmniejszyć** rozmiar tablicy, aby nie zawyżać nadmiernie obszaru niewykorzystanej, lecz zajętej pamięci. Zmniejszanie tablicy jest podobne do jej powiększania: gdy liczba elementów maleje poniżej pewnej granicy, przydzielamy tablicy nowy, mniejszy obszar pamięci i kopiujemy jej zawartość do tego nowego obszaru. Pamięć zajmowana dotychczas przez tablicę zostaje w ten sposób zwolniona i zwrócona do systemu zarządzania pamięcią. Będziemy dążyć do zachowania następujących dwóch własności:

- współczynnik zapełnienia tablicy dynamicznej jest ograniczony z dołu przez pewną stałą,
- koszt zamortyzowany każdej operacji na tablicy jest ograniczony z góry przez stałą.

Przyjmujemy, że koszt można mierzyć za pomocą liczby elementarnych operacji wstawienia lub usunięcia.



Rys. 17.3. Wynik działania ciągu n operacji TABLE-INSERT na liczby: num_i elementów w tablicy, $size_i$ komórek w tablicy oraz potencjał $\Phi_i = 2 \cdot num_i - size_i$, mierzone po wykonaniu i -tej operacji. Cienka linia oznacza num_i , przerywana linia pokazuje $size_i$, a gruba linia – potencjał Φ_i . Zauważmy, że tuż przed powiększeniem tablicy wartość potencjału jest równa liczbie elementów w tablicy, możemy więc za jego pomocą opłacić koszt skopiowania wszystkich elementów do nowej tablicy. Zaraz potem potencjał maleje do 0, lecz jego wartość natychmiast wzrasta o 2 w wyniku wykonania elementarnego wstawienia elementu, który spowodował powiększenie tablicy.

Naturalną strategią powiększania i zmniejszania tablicy jest podwojenie jej rozmiaru, gdy próbujemy wstawić element do wypełnionej tablicy, oraz zmniejszenie rozmiaru o połowę, jeśli usunięcie elementu wywoła zmniejszenie współczynnika wypełnienia poniżej $1/2$. Taka strategia gwarantuje, że współczynnik wypełnienia tablicy nie jest nigdy mniejszy niż $1/2$, lecz niestety dopuszcza dość duży koszt zamortyzowany operacji. Rozważmy następujący scenariusz. Wykonujemy n operacji na tablicy T , gdzie n jest dokładną potęgą 2. Pierwszych $n/2$ operacji stanowią wstawienia, których łączny koszt (zgodnie z naszą wcześniejszą analizą) wynosi $\Theta(n)$. Po wykonaniu tych $n/2$ operacji mamy $num[T] = size[T] = n/2$. Niech pozostałych $n/2$ operacji stanowi następujący ciąg:

I, D, D, I, I, D, D, I, I, ... ,

gdzie I oznacza wstawienie, a D usunięcie elementu. Pierwsze wstawienie powoduje powiększenie tablicy do rozmiaru n . Następne dwie operacje usunięcia powodują zmniejszenie tablicy z powrotem do rozmiaru $n/2$. Dwa kolejne wstawienia ponownie pociągają za sobą powiększenie tablicy itd. Koszt każdego powiększenia i zmniejszenia wynosi $\Theta(n)$. Wykonujemy w sumie $\Theta(n)$ tych operacji, zatem łączny koszt powyższego ciągu n operacji jest równy $\Theta(n^2)$. Stąd koszt zamortyzowany pojedynczej operacji wynosi $\Theta(n)$.

Wada tej strategii jest dość oczywista: po powiększeniu wykonujemy za mało operacji usunięcia, aby zrekompensować koszty zmniejszenia. Podobnie po zmniejszeniu, nie wykonujemy dość wstawień, aby wyrównać wydatki związane z powiększaniem.

Możemy ulepszyć tę strategię kosztem dopuszczenia sytuacji, w której współczynnik zapełnienia maleje poniżej $1/2$. Nadal będziemy podwajać rozmiar tablicy w przypadku, gdy próbujemy wstawić element do pełnej tablicy. Natomiast zmniejszenie rozmiaru o połowę będziemy wykonywać dopiero wtedy, kiedy współczynnik zapełnienia zmaleje poniżej $1/4$, a nie poniżej $1/2$ jak poprzednio. Współczynnik zapełnienia tablicy nadal jest więc ograniczony z dołu przez stałą, lecz teraz będzie to $1/4$. Po wykonaniu powiększenia tablicy jej współczynnik zapełnienia jest równy $1/2$. Zanim będzie można w tej sytuacji wykonać operację zmniejszenia rozmiaru tablicy, trzeba będzie najpierw usunąć połowę jej zawartości, bo zmniejszenie jest dozwolone dopiero wówczas, gdy współczynnik zapełnienia maleje poniżej $1/4$. Po zmniejszeniu rozmiaru tablicy współczynnik zapełnienia jest równy $1/2$. Zanim więc zostanie wykonane powiększenie rozmiaru tablicy, trzeba podwoić liczbę jej elementów, ponieważ operacja powiększania jest dopuszczalna tylko w sytuacji, w której współczynnik zapełnienia przekroczy 1.

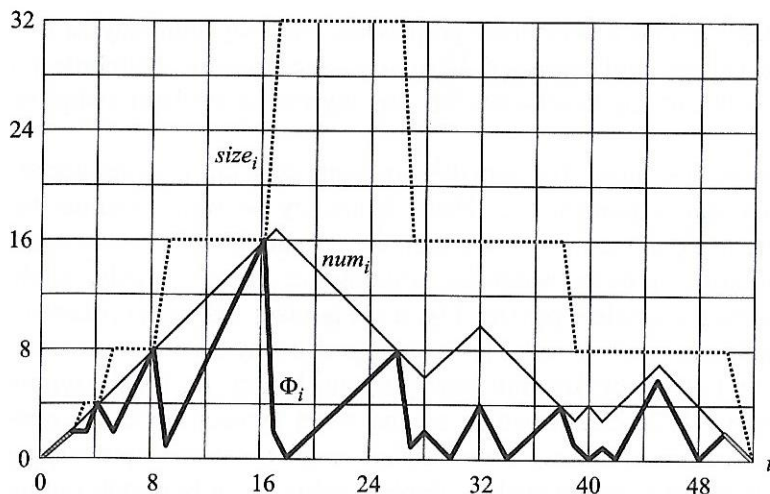
Nie podajemy tutaj implementacji operacji TABLE-DELETE, ponieważ jest ona analogiczna do operacji TABLE-INSERT. Do celów analizy wygodnie będzie przyjąć, że jeśli liczba elementów w tablicy maleje do 0, to pamięć przydzielona tablicy jest zwalniana. Oznacza to, że jeśli $num[T] = 0$, to $size[T] = 0$.

Użyjemy teraz metody potencjału do analizy kosztu ciągu n operacji TABLE-INSERT i TABLE-DELETE. Zdefiniujemy najpierw funkcję potencjału Φ równą 0 zaraz po wykonaniu operacji powiększenia lub zmniejszenia tablicy, a przyrastającą, jeśli współczynnik zapełnienia wzrasta do 1 lub maleje do $1/4$. Oznaczmy współczynnik zapełnienia niepustej tablicy T przez $\alpha(T) = num[T]/size[T]$. Przyjeliśmy, że dla pustej tablicy T mamy $num[T] = size[T] = 0$ oraz $\alpha(T) = 1$, więc dla każdej, także i pustej, tablicy T zachodzi równość $num[T] = \alpha(T) \cdot size[T]$. Zdefiniujemy następująco funkcję potencjału:

$$\Phi(T) = \begin{cases} 2 \cdot num[T] - size[T], & \text{jeśli } \alpha(T) \geq 1/2, \\ size[T]/2 - num[T], & \text{jeśli } \alpha(T) < 1/2. \end{cases} \quad (17.6)$$

Zauważmy, że potencjał pustej tablicy jest równy 0 oraz że potencjał jest zawsze nieujemny. Sumaryczny koszt zamortyzowany ciągu operacji ze względu na Φ stanowi zatem górne ograniczenie ich faktycznego kosztu.

Zanim przejdziemy do ścisłej analizy, przyjrzymy się własnościom funkcji potencjału. Jeżeli współczynnik zapełnienia jest równy $1/2$, to wartość potencjału wynosi 0. Kiedy współczynnik ten osiąga 1, zachodzi równość $size[T] = num[T]$, potencjał jest więc wtedy równy $\Phi(T) = num[T]$, co wystarczy na zapłacenie



Rys. 17.4. Wynik działania ciągu n operacji TABLE-INSERT oraz TABLE-DELETE na liczby: num_i elementów w tablicy, $size_i$ komórek wchodzących w jej skład oraz potencjał Φ_i

$$\Phi_i = \begin{cases} 2 \cdot num_i - size_i & \text{jeśli } \alpha_i \geq 1/2, \\ size_i/2 - num_i & \text{jeśli } \alpha_i < 1/2, \end{cases}$$

mierzone po wykonaniu i -tej operacji. Cienką linią oznaczono num_i , przerywaną – $size_i$, a grubą – potencjał Φ_i . Zauważmy, że tuż przed powiększeniem tablicy wartość potencjału jest równa liczbie elementów w tablicy, możemy więc za jego pomocą opłacić koszt skopiowania wszystkich elementów do nowej tablicy. Podobnie, tuż przed zmniejszeniem tablicy potencjał zrównuje się z liczbą jej elementów.

za koszty powiększenia tablicy, jeśli zajdzie taka potrzeba. Jeśli z kolei współczynnik zapełnienia jest równy $1/4$, to mamy $size[T] = 4 \cdot num[T]$, więc znów $\Phi(T) = num[T]$, czyli wystarczy potencjału na opłacenie wydatków na operację zmniejszenia tablicy. Na rysunku 17.4 widać zmiany potencjału dla pewnego ciągu operacji.

Aby przeanalizować ciąg n operacji TABLE-INSERT i TABLE-DELETE, wprowadźmy następujące oznaczenia: niech c_i oznacza faktyczny koszt i -tej operacji, $\hat{c}_i$ będzie jej kosztem amortyzowanym względem potencjału Φ , num_i niech oznacza liczbę elementów znajdujących się w tablicy po i -tej operacji, $size_i$ – liczbę komórek wchodzących wtedy w skład tablicy, α_i – jej współczynnik zapełnienia po i -tej operacji, a Φ_i – potencjał po i -tej operacji. Początkowo $num_0 = 0$, $size_0 = 0$, $\alpha_0 = 1$ oraz $\Phi_0 = 0$.

Rozważmy najpierw przypadek, w którym i -tą operacją jest TABLE-INSERT. Jeśli $\alpha_{i-1} \geq 1/2$, to analiza przebiega tak samo jak dla powiększania tablicy opisanego w podrozdz. 17.4.1. Niezależnie od tego, czy dochodzi do powiększenia tablicy czy nie, koszt amortyzowany $\hat{c}_i$ operacji nie jest większy niż 3. Jeżeli natomiast $\alpha_{i-1} < 1/2$, to tablica nie może ulec powiększeniu, bo zdarza się to tylko wtedy, gdy $\alpha_{i-1} = 1$. Jeśli zachodzi również $\alpha_i < 1/2$, to koszt amortyzowany

i -tej operacji wynosi

$$\begin{aligned}\hat{c}_i &= c_i + \Phi_i - \Phi_{i-1} \\ &= 1 + (size_i/2 - num_i) - (size_{i-1}/2 - num_{i-1}) \\ &= 1 + (size_i/2 - num_i) - (size_i/2 - (num_i - 1)) \\ &= 0.\end{aligned}$$

Jeśli $\alpha_{i-1} < 1/2$, ale $\alpha_i \geq 1/2$, to

$$\begin{aligned}\hat{c}_i &= c_i + \Phi_i - \Phi_{i-1} \\ &= 1 + (2 \cdot num_i - size_i) - (size_{i-1}/2 - num_{i-1}) \\ &= 1 + (2(num_{i-1} + 1) - size_{i-1}) - (size_{i-1}/2 - num_{i-1}) \\ &= 3 \cdot num_{i-1} - \frac{3}{2}size_{i-1} + 3 \\ &= 3\alpha_{i-1}size_{i-1} - \frac{3}{2}size_{i-1} + 3 \\ &< \frac{3}{2}size_{i-1} - \frac{3}{2}size_{i-1} + 3 \\ &= 3.\end{aligned}$$

Koszt zamortyzowany operacji TABLE-INSERT nie przekracza więc 3.

Zajmijmy się teraz analizą sytuacji, w której i -tą operacją jest TABLE-DELETE. W tym przypadku $num_i = num_{i-1} - 1$. Jeśli $\alpha_{i-1} < 1/2$, to musimy rozważyć, czy operacja powoduje zmniejszenie rozmiaru tablicy. Jeśli nie, to $size_i = size_{i-1}$, a koszt zamortyzowany i -tej operacji wynosi

$$\begin{aligned}\hat{c}_i &= c_i + \Phi_i - \Phi_{i-1} \\ &= 1 + (size_i/2 - num_i) - (size_{i-1}/2 - num_{i-1}) \\ &= 1 + (size_i/2 - num_i) - (size_i/2 - (num_i + 1)) \\ &= 2.\end{aligned}$$

Jeżeli natomiast $\alpha_{i-1} < 1/2$ oraz i -ta operacja powoduje zmniejszenie rozmiaru tablicy, to jej faktyczny koszt wynosi $c_i = num_i + 1$, ponieważ usuwamy jeden oraz kopiujemy num_i elementów. Zachodzą przy tym równości $size_i/2 = size_{i-1}/4 = num_{i-1} = num_i + 1$, wobec tego koszt zamortyzowany i -tej operacji wynosi

$$\begin{aligned}\hat{c}_i &= c_i + \Phi_i - \Phi_{i-1} \\ &= (num_i + 1) + (size_i/2 - num_i) - (size_{i-1}/2 - num_{i-1}) \\ &= (num_i + 1) + ((num_i + 1) - num_i) - ((2 \cdot num_i + 2) - (num_i + 1)) \\ &= 1.\end{aligned}$$

Jeśli i -tą operacją jest TABLE-DELETE, a przy tym $\alpha_{i-1} \geq 1/2$, to koszt zamortyzowany jest także ograniczony przez stałą. Szczegółowa analiza została pozostawiona jako zad. 17.4-2.

Podsumowując, koszt zamortyzowany każdej operacji jest ograniczony przez stałą, faktyczny koszt ciągu n operacji na tablicy dynamicznej wynosi więc $O(n)$.

ZADANIA

- 17.4-1.** Przypuśćmy, że chcemy zaimplementować dynamiczną tablicę z haszowaniem z wykorzystaniem adresowania otwartego. Dlaczego lepiej traktować tablicę jako wypełnioną już wtedy, gdy jej współczynnikapełnienia osiąga pewną ustaloną wartość α mniejszą od 1? Opisz pokrótce, jak zagwarantować, że oczekiwany czas zamortyzowany operacji wstawiania do dynamicznej tablicy z haszowaniem z wykorzystaniem adresowania otwartego jest $O(1)$. Dlaczego wartość oczekiwana faktycznego kosztu operacji wstawiania nie musi być dla wszystkich wstawień równa $O(1)$?
- 17.4-2.** Wykaż, że jeśli i -tą operacją na tablicy dynamicznej jest TABLE-DELETE oraz $\alpha_{i-1} \geq 1/2$, to koszt zamortyzowany tej operacji względem funkcji potencjału (17.6) jest ograniczony przez stałą.
- 17.4-3.** Przypuśćmy, że zamiast dwukrotnego zmniejszania rozmiaru tablicy w sytuacji, gdy jej współczynnikapełnienia maleje poniżej $1/4$, zmniejszamy jej rozmiar, mnożąc go przez $2/3$, jeśli jej współczynnikapełnienia maleje poniżej $1/3$. Użyj funkcji potencjału

$$\Phi(T) = |2 \cdot \text{num}[T] - \text{size}[T]|,$$

by wykazać, że przy użyciu tej strategii koszt zamortyzowany operacji TABLE-DELETE jest ograniczony przez stałą.

Problemy

17-1. Licznik binarny z odwróconymi bitami

W rozdziale 30 omówimy ważny algorytm zwany szybkim przekształceniem Fouriera (FFT, ang. *Fast Fourier Transform*). Pierwszy krok tego algorytmu stanowi wykonanie **permutacji odwracającej bity** na tablicy $A[0..n-1]$ o długości $n = 2^k$ dla pewnej nieujemnej liczby całkowitej k . Polega to na zamianie miejscami elementów, których indeksy zapisane binarnie mają wzajemnie odwrócone bity.

Każdy indeks a można zapisać jako k -bitowy ciąg $\langle a_{k-1}, a_{k-2}, \dots, a_0 \rangle$, gdzie $a = \sum_{i=0}^{k-1} a_i 2^i$. Zdefiniujmy operację rev_k następująco:

$$\text{rev}_k(\langle a_{k-1}, a_{k-2}, \dots, a_0 \rangle) = \langle a_0, a_1, \dots, a_{k-1} \rangle,$$