8.4. Sortowanie kubełkowe

Czas działania **sortowania kubełkowego** w przypadku średnim jest liniowy, przy założeniu że dane wejściowe są wybierane zgodnie z rozkładem jednostajnym. Podobnie jak sortowanie przez zliczanie, sortowanie kubełkowe jest szybkie, ponieważ przyjmuje się tu pewne szczególne założenia dotyczące danych wejściowych. W sortowaniu przez zliczanie przyjmowaliśmy założenie, że elementy do posortowania są małymi liczbami całkowitymi. Przy analizie sortowania kubełkowego będziemy zakładać, że dane wejściowe są liczbami rzeczywistymi wybieranymi losowo z przedziału $[0, 1)$, zgodnie z rozkładem jednostajnym i niezależnie. (Definicja rozkładu jednostajnego znajduje się w dodatku C.2).

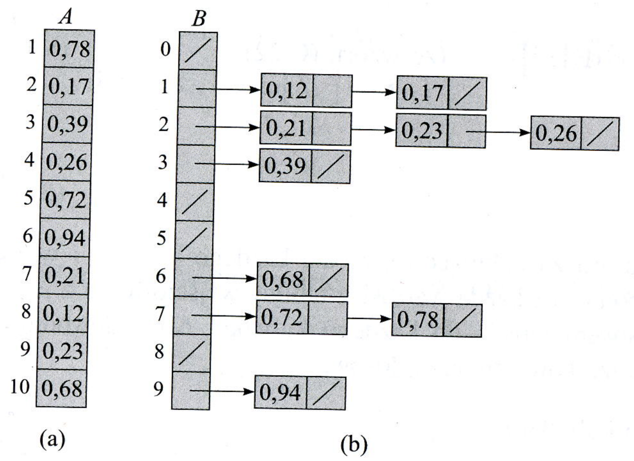
Sortowanie kubełkowe opiera się na triku polegającym na podziale przedziału $[0, 1)$ na n podprzedziałów jednakowych rozmiarów, tzw. **kubełków**, a następnie „rozrzuceniu" n liczb wejściowych do kubełków, do których należą. Ponieważ liczby są rozłożone jednostajnie i niezależnie w przedziale $[0, 1)$, więc oczekujemy, że w każdym z kubełków będzie ich niewiele. Aby otrzymać ciąg wynikowy, sortujemy najpierw liczby w każdym z kubełków, a następnie wypisujemy je, przeglądając po kolei kubełki.

W procedurze BUCKET-SORT przyjmujemy, że dana jest n-elementowa tablica A, a każdy element $A[i]$ w tablicy spełnia warunek $0 \leq A[i] < 1$. W procedurze korzystamy również z pomocniczej tablicy list (kubełków) $B[0\,..\,n-1]$ oraz zakładamy, że są dostępne elementarne operacje na tych listach. (Implementacja elementarnych operacji na stosowanych tu listach z dowiązaniami jest opisana w podrozdz. 10.2).

BUCKET-SORT(A)
$n = A.length$
niech $B[0\,..\,n-1]$ będzie nową tablicą
for $i = 1$ **to** n
    utwórz pustą listę $B[i]$
for $i = 1$ **to** n
    wstaw $A[i]$ na listę $B[\lfloor nA[i] \rfloor]$
for $i = 0$ **to** $n - 1$
    posortuj listę $B[i]$ przez wstawianie
połącz listy $B[0], B[1], \ldots, B[n-1]$ z zachowaniem kolejności

Na rysunku 8.4 jest pokazane działanie sortowania kubełkowego dla tablicy zawierającej 10 liczb.

Zajmiemy się teraz sprawdzeniem poprawności działania sortowania kubełkowego. Rozważmy w tym celu dwa elementy $A[i]$ oraz $A[j]$. Bez straty ogólności możemy założyć, że $A[i] \leq A[j]$. Ponieważ $\lfloor nA[i] \rfloor \leq \lfloor nA[j] \rfloor$, element $A[i]$ zostanie umieszczony albo w tym samym kubełku co $A[j]$, albo w kubełku



Rys. 8.4. Działanie procedury BUCKET-SORT dla $n = 10$. **(a)** Tablica wejściowa $A[1..10]$. **(b)** Tablica $B[0..9]$ posortowanych list (kubełków) po zakończeniu pętli **for** w wierszach 7–8 procedury. Kubełek i zawiera wartości z przedziału $[i/10, (i+1)/10)$. Posortowany ciąg wynikowy powstaje przez połączenie list w kolejności $B[0], B[1], \ldots, B[9]$.

o mniejszym indeksie. Jeśli $A[i]$ i $A[j]$ są umieszczone w tym samym kubełku, to pętla **for** w wierszach 7–8 zapewni ustawienie ich we właściwej kolejności. Jeśli $A[i]$ i $A[j]$ są umieszczone w różnych kubełkach, to wiersz 9 zapewni ustawienie ich we właściwej kolejności. Tak więc algorytm sortowania kubełkowego działa poprawnie.

Przeanalizujemy z kolei czas działania procedury BUCKET-SORT. Zauważmy, że pesymistyczny czas wykonywania wszystkich instrukcji, oprócz wiersza 8, wynosi $O(n)$. Pozostaje więc dokonać analizy czasu potrzebnego do posortowania wszystkich list przez wstawianie w wierszu 8.

Niech n_i będzie zmienną losową oznaczającą liczbę elementów umieszczonych w kubełku $B[i]$. Sortowanie przez wstawianie działa w czasie kwadratowym (patrz podrozdz. 2.2), więc czas działania sortowania kubełkowego wynosi

$$T(n) = \Theta(n) + \sum_{i=0}^{n-1} O(n_i^2).$$

Przeanalizujemy teraz czas działania sortowania kubełkowego w przypadku średnim, obliczając wartość oczekiwaną czasu działania ze względu na rozkład danych wejściowych. Biorąc wartości oczekiwane obu stron i korzystając z liniowości wartości oczekiwanej, mamy

$$\begin{aligned} \mathrm{E}[T(n)] &= \mathrm{E}\left[\Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)\right] \\ &= \Theta(n) + \sum_{i=0}^{n-1} \mathrm{E}\left[O(n_i^2)\right] \qquad \text{(z liniowości wartości oczekiwanej)} \end{aligned}$$

$$= \Theta(n) + \sum_{i=0}^{n-1} O(\mathrm{E}\left[n_i^2\right]) \qquad \text{(ze wzoru (C.22))}. \tag{8.1}$$

Twierdzimy, że

$$\mathrm{E}\left[n_i^2\right] = 2 - 1/n \tag{8.2}$$

dla $i = 0, 1, \ldots, n-1$. Nie jest zaskakujące to, że dla każdego kubełka i wartość $\mathrm{E}\left[n_i^2\right]$ jest taka sama, ponieważ każda wartość z tablicy wejściowej A z jednakowym prawdopodobieństwem wpada do każdego kubełka. Aby udowodnić wzór (8.2), zdefiniujmy wskaźnikowe zmienne losowe

$$X_{ij} = \mathrm{I}\{A[j] \text{ wpada do kubełka } i\}$$

dla $i = 0, 1, \ldots, n-1$ oraz $j = 1, 2, \ldots, n$. Mamy zatem

$$n_i = \sum_{j=1}^{n} X_{ij}.$$

Żeby obliczyć $\mathrm{E}\left[n_i^2\right]$, podnosimy do kwadratu i przegrupowujemy składniki:

$$\begin{aligned}
\mathrm{E}\left[n_i^2\right] &= \mathrm{E}\left[\left(\sum_{j=1}^{n} X_{ij}\right)^2\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n}\sum_{k=1}^{n} X_{ij}X_{ik}\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n} X_{ij}^2 + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n \\ k\ne j}} X_{ij}X_{ik}\right] \\
&= \sum_{j=1}^{n} \mathrm{E}\left[X_{ij}^2\right] + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n \\ k\ne j}} \mathrm{E}\left[X_{ij}X_{ik}\right],
\end{aligned} \tag{8.3}$$

gdzie ostatnia równość wynika z liniowości wartości oczekiwanej. Te dwie sumy obliczamy osobno. Wskaźnikowa zmienna losowa X_{ij} jest równa 1 z prawdopodobieństwem $1/n$, a 0 w przeciwnym razie, zatem

$$\begin{aligned}
\mathrm{E}\left[X_{ij}^2\right] &= 1^2 \cdot \frac{1}{n} + 0^2 \cdot \left(1 - \frac{1}{n}\right) \\
&= \frac{1}{n}.
\end{aligned}$$

Dla $k \neq j$ zmienne X_{ij} i X_{ik} są niezależne, zatem

$$\begin{aligned} \mathrm{E}\left[X_{ij}X_{ik}\right] &= \mathrm{E}\left[X_{ij}\right]\mathrm{E}\left[X_{ik}\right] \\ &= \frac{1}{n}\cdot\frac{1}{n} \\ &= \frac{1}{n^2}. \end{aligned}$$

Wstawiając te dwie wartości oczekiwane do równania (8.3), dostajemy

$$\begin{aligned} \mathrm{E}\left[n_i^2\right] &= \sum_{j=1}^{n}\frac{1}{n} + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n \\ k\neq j}}\frac{1}{n^2} \\ &= n\cdot\frac{1}{n} + n(n-1)\cdot\frac{1}{n^2} \\ &= 1 + \frac{n-1}{n} \\ &= 2 - \frac{1}{n}, \end{aligned}$$

co kończy dowód wzoru (8.2).

Wstawiając tę wartość oczekiwaną do równania (8.1), wnioskujemy, że w przypadku średnim czas działania sortowania kubełkowego wynosi $\Theta(n) + n \cdot O(2 - 1/n) = \Theta(n)$.

Nawet jeśli dane wejściowe nie są wybierane zgodnie z rozkładem jednostajnym, sortowanie kubełkowe może i tak działać w czasie liniowym. Jeśli tylko dane wejściowe mają tę własność, że suma kwadratów rozmiarów kubełków jest liniowa względem łącznej liczby elementów, to z równania (8.1) wynika, że sortowanie kubełkowe będzie działało w czasie liniowym.

ZADANIA

8.4-1. Zilustruj (podobnie jak na rys. 8.4) działanie procedury BUCKET-SORT dla tablicy $A = \langle 0{,}79,\ 0{,}13,\ 0{,}16,\ 0{,}64,\ 0{,}39,\ 0{,}20,\ 0{,}89,\ 0{,}53,\ 0{,}71,\ 0{,}42\rangle$.

8.4-2. Wyjaśnij, dlaczego pesymistyczny czas działania algorytmu sortowania kubełkowego to $\Theta(n^2)$. Jak w prosty sposób zmodyfikować ten algorytm, aby jego czas działania w przypadku średnim był nadal liniowy, a czas pesymistyczny wynosił $O(n \lg n)$?

8.4-3. Niech X będzie zmienną losową określającą liczbę orłów uzyskanych w wyniku dwukrotnego rzutu symetryczną monetą. Jaka jest wartość oczekiwana kwadratu zmiennej losowej, czyli $\mathrm{E}[X^2]$, a jaki jest kwadrat jej wartości oczekiwanej, czyli $\mathrm{E}^2[X]$?

★ **8.4-4.** Danych jest n punktów $p_i = (x_i, y_i)$ leżących w kole jednostkowym, tzn. spełniających warunek $0 < x_i^2 + y_i^2 \le 1$ dla $i = 1, 2, \ldots, n$. Załóżmy, że