ćwiczenia (licencjat wieczorowy): lista zadań nr 8 *27 listopada 2007 r.*

ALGORYTMY I STRUKTURY DANYCH

HASZOWANIE

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

1. [∗] Wykaż, że jeśli uniwersum U jest wystarczająco duże $|U| \geq nm$, to istnieje podzbiór $K \subset U$ o rozmiarze $|K| = n$, składający się z kluczy które są odwzorowywane za pomocą funkcji haszującej h na tę samą pozycję w tablicy z haszowaniem $T[0 \ldots m-1]$. Jaki jest w związku z tym pesymistyczny czas wyszukiwania elementu w tablicy z haszowaniem z *łańcuchową metodą rozwiązywania konfliktów*?

2. [∗] Dana jest tablica $T[0 \ldots m-1]$ z funkcją haszującą $h(k) = \lfloor m(k\alpha \bmod 1) \rfloor$, gdzie $m = 1000$ i $\alpha = \frac{\sqrt{5}-1}{2}$. Oblicz pozycje, na które trafią klucze 1, 2, ..., 16.

   Uwaga. Do rozwiązania tego zadania wykorzystaj komputer.

3. [∗] Do 7–elementowej początkowo pustej *tablicy z haszowaniem* $T[0 \ldots 6]$ wstawiamy kolejno elementy: 11, 19, 53, 49, 41, 23 i 61. Powstające konflikty rozwiązujemy metodą *adresowania otwartego liniowego*. Ile konfliktów wystąpi przy wstawianiu do niej podanych wartości? Następnie z tablicy usuwamy elementy o wartościach 19, 53 i 41. Zilustruj przeprowadzane na tej tablicy operacje i przedstaw ostateczną zawartość tablicy T.

4. [∗] Do 8–elementowej początkowo pustej *tablicy z haszowaniem* $T[0 \ldots 7]$ wstawiamy kolejno elementy: 29, 87, 43, 11 i 59. Powstające konflikty rozwiązujemy metodą *adresowania otwartego kwadratowego*; jako funkcję haszującą stosujemy: $h'(k) = k \bmod 8$ oraz współczynniki $a = 4$ i $b = 3$. Oblicz, na jakie pozycje trafią wstawiane wartości. Zilustruj przeprowadzane na tej tablicy operacje i przedstaw ostateczną zawartość tablicy T.

5. [∗] Do 7–elementowej początkowo pustej *tablicy z haszowaniem* $T[0 \ldots 6]$ wstawiamy kolejno elementy: 11, 23, 37, 49, 57, 65 i 73. Powstające konflikty rozwiązujemy metodą *adresowania otwartego z podwójnym haszowaniem*; jako funkcje haszujące stosujemy: $h'(k) = k \bmod 7$ oraz $h''(k) = 1 + (k \bmod 6)$. Oblicz, na jakie pozycje trafią wstawiane wartości. Zilustruj przeprowadzane na tej tablicy operacje i przedstaw ostateczną zawartość tablicy T.

6. [∗∗] Zaimplementuj w pseudokodzie procedurę wstawiającą *insert* i usuwającą *delete* zadaną wartość do/z tablicy z haszowaniem z *adresowaniem otwartym*.

7. [∗∗] Zaproponuj metodę implementacji tablicy z *adresowaniem bezpośrednim*, w której klucze pamiętanych elementów nie muszą być różne, a za to z każdym elementem mogą być związane pewne dodatkowe informacje. Wiemy też, że elementów o takim samym kluczu może być co najwyżej kilka $O(1)$. Wszystkie operacje słownikowe powinny działać w czasie stałym $O(1)$.

8. [∗∗∗] Chcemy zaimplementować słownik za pomocą *adresowania bezpośredniego* w bardzo dużej tablicy (tablica ze wskaźkikami). Początkowo w tablicy tej mogą się znajdować przypadkowe wartości, ale jej zainicjalizowanie nie wchodzi w rachubę ze względu na olbrzymi rozmiar tablicy. Zaprojektuj metodę reprezentowania dużego słownika w tablicy z adresowaniem bezpośrednim bez konieczności inicjalizowania wszystkich slotów. Każdy przechowywany element powinien zajmować $O(1)$ komórek pamięci. Operacje słownikowe *insert*, *delete* i *search* powinny działać w stałym czasie $O(1)$. Również przygotowanie struktury danych do użytkowania powinno być wykonane w czasie stałym.

   Wskazówka. Użyj dodatkowego stosu o rozmiarze równym liczbie elementów w tablicy, za pomocą którego można będzie określić, czy dana pozycja w tablicy zawiera sensowną wartość.