ćwiczenia (licencjat dzienny): lista zadań nr 8 *28 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. [$***$] Niech T będzie tablicą z haszowaniem o rozmiarze $|T| = m$, gdzie m jest ustaloną liczbą pierwszą. Klucze, które będziemy zapisywać w tej tablicy, to $(r+1)$–elementowe ciągi znaków $k = \langle k_0, k_1, \ldots, k_r \rangle$, przy czym znaki pochodzą ze zbioru $Z = \{0, 1, \ldots, m-1\}$ (maksymalna wartość znaku jest $< m$). Niech $a = \langle a_0, a_1, \ldots, a_r \rangle$ będzie $(r+1)$–elementowym ciągiem znaków wybranych losowo ze zbioru Z. Odpowiadająca temu ciągowi funkcja haszująca h_a jest zdefiniowana następująco: $h_a(x) = (\sum_{i=0}^{r} a_i x_i) \bmod m$. Wykaż, że rodzina funkcji $\mathcal{H} = \bigcup_{a \in Z^{r+1}} h_a$ jest uniwersalną rodziną funkcji haszujących.

3. [$**$] Załóżmy, że klucze są t-bitowymi liczbami całkowitymi. Udowodnij, że dla modularnej funkcji haszującej $h(k) = k \bmod m$, gdzie $m > 2$ jest liczbą pierwszą, prawdziwa jest następująca własność: każde dwa klucze k_i i k_j różniące się tylko na jednym bicie mają różne wartości funkcji haszującej $h(k_i) \neq h(k_j)$.

4. [$***$] 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.

5. [$***$] Wykaż, że w tablicy z haszowaniem z *łańcuchową metodą rozwiązywania konfliktów* prawdopodobieństwo, że liczba kluczy na liście wynosi około $\frac{n}{m}$ jest bliskie 1 (m to rozmiar tablicy a n to liczba pamiętanych w niej elementów).

   Wskazówka. Wylicz, jakie jest prawdopodobieństwo, że lista będzie miała dokładnie k elementów.

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

7. [$***$] Niech dana będzie tablica z haszowaniem o rozmiarze 2^r dla $r \in \mathbf{N}$, w której konflikty są usuwane za pomocą *adresowania otwartego kwadratowego.* Wykaż, że w takim przypadku funkcja $h(k, i) = (k + 2i^2 + i) \bmod 2^r$ trafia we wszystkie pozycje w tablicy dla $i = 0, \ldots, 2^r - 1$. Czy własność ta będzie nadal zachowana, gdy współczynnik przy i zmienimy na dowolną większą liczbę nieparzystą lub współczynnik przy i^2 na dowolną większą liczbę parzystą? Zakładamy, że oba współczynniki są $< 2^r$.

8. [$**$] Załóżmy, że zaimplementowałeś tablicę z haszowaniem, w której konflikty są usuwane za pomocą *adresowania otwartego z podwójnym haszowaniem.* Programując funkcje haszujące popełniłeś taki błąd, że jedna lub druga funkcja zawsze zwraca tę samą wartość $\neq 0$. Opisz co się zdarzy w sytuacji gdy: *(i)* pierwsza funkcja jest błędna, *(ii)* druga funkcja jest błędna, *(iii)* obie funkcje są błędne.