ASD - ćwiczenia X

Kolejki priorytetowe

- operacje kolejki priorytetowej:
  - $EMPTY\ :\ \mathcal{PQ} \to \{TURE, FALSE\}$, sprawdzenie czy struktura jest pusta,
  - $INSERT\ :\ \mathcal{PQ} \times E \to PQ$, wstawienie elementu do struktury struktury,
  - $MAX\ :\ \mathcal{PQ} \to E$, „obejrzenie“ maksymalnego elementu przechowywanego w strukturze,
  - $DELMAX\ :\ \mathcal{PQ} \to \mathcal{PQ}$, usunięcie maksymalnego elementu przechowywanego w strukturze,

  gdzie $\mathcal{PQ}$ jest przestrzenią kolejek priorytetowych, E przestrzenią elementów przechowywanych w kolejce priorytetowej,
- złożoność czasowa operacji n-elementowej kolejki priorytetowej w przypadku implementacji struktury w kopcu typu max:
  - $A\left(\texttt{EMPTY}(), n\right) = W\left(\texttt{EMPTY}(), n\right) = O\left(1\right)$,
  - $A\left(\texttt{EMPTY}(), n\right) = W\left(\texttt{INSERT}(), n\right) = O\left(\log\left(n\right)\right)$,
  - $A\left(\texttt{EMPTY}(), n\right) = W\left(\texttt{MAX}(), n\right) = O\left(1\right)$,
  - $A\left(\texttt{EMPTY}(), n\right) = W\left(\texttt{DELMAX}(), n\right) = O\left(\log\left(n\right)\right)$,

Słowniki

- operacje słownikowe:
  - $EMPTY\ :\ \mathcal{D} \to \{TURE, FALSE\}$, sprawdzenie czy struktura jest pusta,
  - $INSERT\ :\ \mathcal{D} \times E \to \mathcal{D}$, wstawienie elementu do struktury,
  - $DELETE\ :\ \mathcal{D} \times E \to \mathcal{D}$, usunięcie elementu ze struktury,
  - $MEMBER\ :\ \mathcal{D} \times E \to \{TURE, FALSE\}$, sprawdzenie, czy dany element jest przechowywany w strukturze,

  gdzie $\mathcal{D}$ jest przestrzenią słowników, E przestrzenią elementów przechowywanych w słowniku,
- złożoność czasowa operacji n-elementowego słownika w przypadku implementacji struktury w drzewie typu AVL:
  - $A\left(\texttt{EMPTY}(), n\right) = O\left(1\right)$, $W\left(\texttt{EMPTY}(), n\right) = O\left(1\right)$,
  - $A\left(\texttt{INSERT}(), n\right) = O\left(\log\left(n\right)\right)$, $W\left(\texttt{INSERT}(), n\right) = O\left(\log\left(n\right)\right)$,
  - $A\left(\texttt{DELETE}(), n\right) = O\left(\log\left(n\right)\right)$, $W\left(\texttt{DELETE}(), n\right) = O\left(\log\left(n\right)\right)$,
  - $A\left(\texttt{MEMBER}(), n\right) = O\left(\log\left(n\right)\right)$, $W\left(\texttt{MEMBER}(), n\right) = O\left(\log\left(n\right)\right)$.
- złożoność czasowa operacji n-elementowego słownika w przypadku implementacji struktury w k-elementowej tablicy mieszającej (z dodatkową zmienną zliczającą) z łańcuchową metodą rozwiązywania kolizji:
  - $A\left(\texttt{EMPTY}(), n, k\right) = O\left(1\right)$, $W\left(\texttt{EMPTY}(), n, k\right) = O\left(1\right)$,
  - $A\left(\texttt{INSERT}(), n, k\right) = O\left(\frac{n}{k}\right)$, $W\left(\texttt{INSERT}(), n, k\right) = O\left(n\right)$,
  - $A\left(\texttt{DELETE}(), n, k\right) = O\left(\frac{n}{k}\right)$, $W\left(\texttt{DELETE}(), n, k\right) = O\left(n\right)$,
  - $A\left(\texttt{MEMBER}(), n, k\right) = O\left(\frac{n}{k}\right)$, $W\left(\texttt{MEMBER}(), n, k\right) = O\left(n\right)$.

Zadania

1. Dany jest kopiec H o n elementach, gdzie $n > 1$, którego korzeń zawiera element maksymalny. Jaki jest koszt znalezienia k-tego co do wielkości elementu w kopcu H, jeżeli wolno stosować tylko operacje $INSERT$, $DELMAX$, MAX, $EMPTY$ (charakterystyczne dla kolejek priorytetowych)? Zakładamy, że element ten istnieje.

2. W pewnym kinie znajduje się specjalna sala projekcyjna dla VIP-ów, w której zainstalowano tylko jeden rząd $n > 1$ foteli ponumerowanych liczbami od 1 do n. Przy zakupie biletów na seanse filmowe w tej sali, klienci kina postępują zgodnie z poniższymi regułami:

   - pierwszy klient kupuje bilet na miejsce z numerem k,
   - każdy kolejny klient kupuje bilet na miejsce, które jest maksymalnie oddalone od każdego z już zajętych foteli. Jeżeli więcej niż jedno miejsce spełnia ten warunek, to klienci wybierają miejsce bliższe wejścia do sali (tj. miejsce o niższym numerze).

   Jesteś ostatnim klientem, któremu udało się zakupić bilet na seans w sali dla VIP-ów. Na którym fotelu będziesz siedział w czasie projekcji filmu?

   (a) Zaprojektuj strukturę danych i funkcję

   `int SEAT(int n, int k),`

   która pozwoli w maksymalnie efektywny sposób odpowiedzieć na zadane pytanie.

   (b) Oszacuj złożoność czasową funkcji `SEAT()` względem liczby n.

3. Niech słowo a należy do pewnego języka A, wtedy koszt operacji $FIND(D, a)$ dla haszowania modularnego z łańcuchową metodą rozwiązywania problemu kolizji w przypadku pesymistycznym wynosi $O(k)$, gdzie k jest liczbą słów z języka A zapisanych w słowniku D (przyjmujemy, że operacją dominującą jest porównywanie słowa a z dowolnym słowem ze słownika D). Zmodyfikuj metodę rozwiązywania kolizji tak, aby złożoność operacji wyszukiwania w słowniku D wynosiła w przypadku pesymistycznym $O(\log(k))$, przy zachowaniu warunku jednokrotnego wyliczania wartości funkcji haszującej $f(a)$. Określ złożoność rozważanej operacji słownikowej (dla zmodyfikowanej metody rozwiązywania kolizji) w przypadku oczekiwanym (średnim).

4. Zaprojektuj strukturę danych typu $\mathcal{S}$ nad zbiorem liczb naturalnych $\mathbb{N}$, pozwalającą na wykonanie następujących operacji w czasie $O(\log(n))$:

   - $EMPTY : \mathcal{S} \rightarrow \{TURE, FALSE\}$, sprawdzenie czy struktura jest pusta,
   - $INSERT : \mathcal{S} \times \mathbb{N} \rightarrow \mathcal{S}$, wstawienie liczby naturalnej do struktury,
   - $DELETE : \mathcal{S} \times \mathbb{N} \rightarrow \mathcal{S}$, usunięcie liczby naturalnej ze struktury,
   - $MEMBER : \mathcal{S} \times \mathbb{N} \rightarrow \{TURE, FALSE\}$, sprawdzenie, czy dana liczba naturalna jest przechowywany w strukturze,
   - $PRED\_SUM : \mathcal{S} \times \mathbb{N} \rightarrow \mathbb{N}$, podanie sumy elementów mniejszych od rozważanego,
   - $SUCC\_SUM : \mathcal{S} \times \mathbb{N} \rightarrow \mathbb{N}$, podanie sumy elementów większych od rozważanego.

5. Podaj zawartość tablicy mieszającej (Hash Table) po wstawieniu elementów

E, A, S, Y, Q, U, T, I, O, N, S

do początkowo pustej 5-elementowej tablicy M. Do zakodowania i-tej litery alfabetu angielskiego użyj funkcji mieszającej $\beta : \mathbb{N} \to \mathbb{N}$ postaci $\beta(i) = 11i \bmod 5$, np.

$$\begin{aligned} \beta(index(F)) &= \beta(6) \\ &= 11 \cdot 6 \bmod 5 \\ &= 1. \end{aligned}$$