Algorytmy i Struktury Danych

1. (0pkt) Przeczytaj treści zadań z tej listy. Jeśli zauważysz, że nie rozumiesz treści co najmniej trzech zadań, wyciągnij wniosek, że jest kiepsko.

2. (1pkt) Czy następujące stwierdzenie jest prawdziwe: Jeśli operację wstawiania elementu do 2-3-4 drzewa dokonujemy metodą zstępującą, to żaden 3-wierzchołek nie ma wśród swoich dzieci 3-wierzchołka. Przypomnienie terminologii: 2-3-4 drzewa, to B-drzewa, których wierzchołki mogą pamiętać od 1 do 3 kluczy; 3-wierzchołek, to wierzchołek pamiętający 3 klucze.

3. (2pkt) Niech $A = \{a_1, a_2, \ldots, a_n\}$ będzie zbiorem kluczy, (takim, że $\forall_{i=1,\ldots,n-1} a_i < a_{i+1}$), które chcemy pamiętać w słowniku stałym. Znamy także ciąg $p_1, \ldots, p_n$ prawdopodobieństw zapytania o poszczególne klucze. Przyjmujemy, że $p_1 + p_2 + \ldots + p_n = 1$, a więc do słownika nie będą kierowane zapytania o klucze spoza słownika. Chcemy zaimplementować słownik jako drzewo BST. Ułóż algorytm znajdujący takie drzewo, które minimalizuje oczekiwany czas wykonywania operacji na słowniku.

   Uwaga: w grupie zaawansowanej należy podać algorytm działający w czasie $O(n^2)$; w pozostałych grupach wystarczy podać algorytm działający w czasie $O(n^3)$.

4. (1pkt) Uzasadnij stwierdzenie, że funkcje hashujące potrzebne w konstrukcji słownika stałego (podanej na wykładzie) mogą być efektywnie znalezione.

5. (2pkt) Przeprowadź analizę oczekiwanego czasu wykonywania operacji *insert* w losowym drzewcu.

6. (1pkt) Rodzinę H funkcji hashujących ze zbioru kluczy U w zbiór indeksów $\{0, .., m-1\}$ nazywamy *uniwersalną*, jeśli

$$\forall_{x \neq y \in U} Pr_{h \in H}[h(x) = h(y)] \leq \frac{1}{m}$$

   Podaj przykład uniwersalnej rodziny funkcji hashujących.