

Zadanie «Złożoność»

nr IKU _____

liczba punktów _____ / 15

Za zadanie można otrzymać 15 punktów.

Dana jest tablica tab zawierająca n (gdzie $n = 2^k - 1$; $k = 1, 2, \dots$) uporządkowanych rosnąco kluczy. Klucze w tablicy są indeksowane od 1 do n .

Dana jest rekurencyjna funkcja WYSZUKIWANIEBINARNE implementująca algorytm wyszukiwania binarnego klucza x w tablicy tab , gdzie $lewy$ jest indeksem skrajnie lewego elementu, a $prawy$ jest indeksem skrajnie prawego elementu w przeszukiwanej tablicy. Funkcja jako wynik zwraca indeks pozycji w tablicy tab na której znajduje się szukany klucz x , lub wartość -1 jeżeli tablica tab nie zawiera klucza x . Treść funkcji jest następująca:

```
1: function WYSZUKIWANIEBINARNE( $tab, lewy, prawy, x$ )
2:    $i := (lewy + prawy) \text{ div } 2$ ;
3:   if  $tab[i] = x$  then                                     ▷ znaleziono klucz  $x$  w tablicy  $tab$  na pozycji  $i$ 
4:     return  $i$ ;
5:   else if  $lewy = prawy$  then                               ▷ tablica  $tab$  nie zawiera klucza  $x$ 
6:     return  $-1$ ;
7:   else if  $tab[i] > x$  then                                  ▷ poszukiwanie klucza  $x$  w lewym fragmencie tablicy
8:     return WYSZUKIWANIEBINARNE( $tab, lewy, i - 1, x$ );
9:   else                                                     ▷ poszukiwanie klucza  $x$  w prawym fragmencie tablicy
10:    return WYSZUKIWANIEBINARNE( $tab, i + 1, prawy, x$ );
11:  end if
12: end function
```

Funkcję wywołano w następujący sposób:

$$\text{WYSZUKIWANIEBINARNE}(tab, 1, 31, x);$$

przekazując do niej tablicę tab zawierającą 31 kluczy (przyjąć, że tablica tab zawiera szukany klucz x). Klucz x może znajdować się w tablicy na jednej z 31 pozycji z jednakowym prawdopodobieństwem. Dla powyższego wywołania funkcji należy wykonać:

1. Wyszukiwanie klucza x w tablicy tab można opisać za pomocą drzewa binarnego. Węzły w takim drzewie zawierają indeksy kluczy w tablicy tab , które są porównywane w kolejnych iteracjach z poszukiwanym kluczem x (korzeń drzewa zawiera indeks klucza, który zostanie porównany jako pierwszy, węzły potomne korzenia zawierają indeksy porównywanych kluczy jako drugie, itd.). Należy narysować pełne drzewo binarne opisujące wyszukiwanie klucza x w tablicy tab mając na uwadze to, że szukany klucz może znajdować się na dowolnej pozycji w tablicy tab . (3 punkty)
2. Wyznaczyć liczbę porównań klucza x z kluczami w tablicy tab w przypadku pesymistycznym (porównania wykonywane w wierszach 3 i 7 funkcji). (3 punkty)

3. Wyznaczyć średnią liczbę porównań klucza x z kluczami w tablicy tab (porównania wykonywane w wierszach 3 i 7 funkcji). **(5 punktów)**
4. Wyznaczyć prawdopodobieństwo, że podczas wyszukiwania zostanie wykonanych nie więcej niż 5 porównań klucza x z kluczami znajdującymi się w tablicy tab (porównania wykonywane w wierszach 3 i 7 funkcji). **(4 punkty)**

W punktach 2, 3, 4 jako wynik należy przedstawić odpowiednie obliczenia prowadzące do końcowego wyniku, a także krótko skomentować co przedstawiają obliczenia.