

Teoretyczne podstawy informatyki
Zestaw zadań nr 6
Wszystkie grupy

dr Anna Ochab-Marcinek

1. Dla danej listy $L = (2,7,1,8,2)$ określ: jaka jest jej długość, nagłówek, stopka, podaj wszystkie przedrostki, wszystkie przyrostki, wszystkie podlisty. Patrz [1], rozdział 6.2.
2. Co robią poniższe funkcje? Ile czasu wymaga ich wykonanie, w zależności od długości listy? Patrz [1], rozdział 6.4.

```
#include <stdio.h>

#define TRUE 1
#define FALSE 0

typedef int BOOLEAN;

typedef struct CELL *LIST;
struct CELL {
    int element;
    LIST next;
};

BOOLEAN lookup(int x, LIST L)
{
    if (L == NULL)
        return FALSE;
    else if (x == L->element)
        return TRUE;
    else
        return lookup(x, L->next);
}

void delete(int x, LIST *pL)
```

```

{
    if ((*pL) != NULL)
        if (x == (*pL)->element)
            (*pL) = (*pL)->next;
        else
            delete(x, &((*pL)->next));
}

void insert(int x, LIST *pL)
{
    if ((*pL) == NULL) {
        (*pL) = (LIST) malloc(sizeof(struct CELL));
        (*pL)->element = x;
        (*pL)->next = NULL;
    }
    else if (x != (*pL)->element)
        insert(x, &((*pL)->next));
}

```

3. Model danych oparty na drzewach. Co oznaczają poniższe terminy? Węzeł, krawędź, korzeń drzewa, rodzic, dziecko, drzewo spójne, ścieżka w drzewie, długość ścieżki, poddrzewo, liść, węzeł wewnętrzny, wysokość drzewa, głębokość drzewa. Proszę pokazać to na przykładowych rysunkach.
4. Na czym polega reprezentacja tablicowa, a na czym reprezentacja *lewy potomek, prawy element siostrzany*? Jak zaimplementować je w języku C? Patrz [1], rozdział 5.3.
5. Jaka jest różnica między drzewami zwykłymi a drzewami binarnymi? Jak zaimplementować je w języku C? Patrz [1], rozdział 5.6.
6. [1], rozdział 5.7, przykład 5.22 i 5.23. Wytłumaczyć, co to jest drzewo przeszukiwania binarnego i jak działa wyszukiwanie danego słowa w takim drzewie. Poniżej listing z przykładu 5.23.

```

#include <stdio.h>

#define TRUE 1
#define FALSE 0

typedef int BOOLEAN;
typedef int ETYPE;

typedef struct NODE *TREE;

```

```
struct NODE {
    ETYPE element;
    TREE leftChild, rightChild;
};

BOOLEAN lookup(ETYPE x, TREE T)
{
    if (T == NULL)
        return FALSE;
    else if (x == T->element)
        return TRUE;
    else if (x < T->element)
        return lookup(x, T->leftChild);
    else /* x must be > T->element */
        return lookup(x, T->rightChild);
}
```

Literatura

- [1] A.V. Aho, J.D. Ullman, *Wykłady z informatyki z przykładami w języku C*