

# Algoritmusok 1

## 6. csoport 1. ZH

Minden feladat 10 pontot ér, a sikeres zh-hoz legalább 20 pontot kell elérni.

A struktogramokhoz a fejlécüket is fel kell írni, típusokkal együtt. A referenciaként átadott paramétereket jelöljük a típus végére írt „&” jellel.

- 1) A tanult algoritmust használva készítsd el az alábbi kifejezés lengyel formáját! A többszörös hatványozást jobbról balra kell végrehajtani! Ábrázold követhető módon a verem tartalmát! (Precedencia: = 0, +- 1, \*/ 2, ^ 3)  
a)  $x = a + b * (c + d) + e * (f / g) + a - b^{2^2}$
- 2) Növekvő sorrendbe kell rendezni a következő egész tömböt: [5, 9, 3, 1, 6, 8, 5, 4, 2].  
a) Mutasd be, hogyan működne a tanult Merge sort (Összefésülő rendezés) algoritmus a tömbön! Ábrázold a szétvágásokat, majd az összefuttatásokat. Ha páratlan számú elemet kell szétvágni, akkor balra jusson kevesebb elem.  
b) Adj aszimptotikusan éles korlátot a merge sort futásidejére  $n$  hosszú tömb esetén.  $\Theta(???)$  Mennyi a gyorsrendezés futásideje legrosszabb esetben,  $n$  hosszú tömb esetén?  $\Theta(???)$

- 3) Készítsd el a következő program struktogramját:  
S sorban adott egy 1-nél nagyobb egész szám prímtényező felbontása. (Esetleg lehet üres a sor.) A prímtényezők növekvően vannak a sorban, lásd a példákat. T sorban állítsuk elő azt a számot, melynek prímtényező felbontása az S-ben felsorolt prímtényezők közül mindegyiket pontosan egyszer tartalmazza. A futásidő legyen  $O(n)$ , ha  $n = L(S)$ .

S = 2, 2, 2, 5, 7, 7, 13  
S = 5  
S = (üres)

T = 2, 5, 7, 13  
T = 5  
T = (üres)

- 4) Készítsd el a következő program struktogramját:  
Az **L** nevű fejelemes egyirányú esetleg üres (H1L) lista egész számokat tartalmaz **csökkenő** sorrendben. Az **r** pointer egy új listaelemre mutat, ami szintén egy egész számot tartalmaz. Szűrjük be úgy a listába, hogy a rendezettség megmaradjon. Egyéb adatszerkezetet ne használjunk, a futásidő legyen  $O(n)$ , ha a lista hossza  $n$ .

- 5) Adott két kétirányú fejelemes ciklikus (C2L) lista: L1 és L2. A listák növekvően rendezettek, egy-egy 1-nél nagyobb egész szám prímtényező felbontását tartalmazzák. Készíts LKKT(L1, L2) néven algoritmust, mely egy L2 lista megfelelő elemeinek átfűzésével előállítja L1-ben a két szám legkisebb közös többszörösének prímtényező felbontását. (Ha egy prímtényező kezdetben L1-ben  $a$ -szor szerepelt, L2-ben  $b$ -szer, akkor az eredményben  $\max(a, b)$ -szer kell szerepelnie.) A futásidő legyen  $O(n)$ . Egyéb adatszerkezetet ne használjunk. A tanult follow, precede és out műveletek használhatóak, nem kell felírni a definíciójukat.

Példa1: L1 = 2, 2, 2, 3    L2 = 2, 3, 5  $\rightarrow$  L1 = 2, 2, 2, 3, 5  
Példa2: L1 = 3, 7, 7    L2 = 2, 2, 7  $\rightarrow$  L1 = 2, 2, 3, 7, 7

- 6) Írj struktogramot, ami egy +-\*/ kétváltozós műveletekből és egyjegyű számokból álló sorozatról eldönti, hogy helyes lengyel forma-e. A lengyel formát a standard bemenetről olvassuk, az eredményt (True/False) a standard kimenetre írjuk. Az üres string helyes.  
Példa1: 1 2 + \*  $\rightarrow$  False,    Példa2: + 1  $\rightarrow$  False  
Példa3: 1 2 + 3 \*  $\rightarrow$  True