

Témakörök:**Lassú rendezések tömbre (algoritmust nem kell leírni, csak bemutatni a működésüket)**

- Buborék, javított buborék, maximum/minimum kiválasztó, beszűrő rendezés

Gyors rendezések tömbre (algoritmust nem kell leírni, csak bemutatni a működésüket)

- összefésülő rendezés (mergesort), gyorsrendezés (quicksort)

Lengyel formára hozás, lengyel forma kiértékelés

- aritmetikai műveleteket tartalmazó, helyesen zárójelezett kifejezésen a tanult algoritmus lejátszása, a verem tartalmának nyomon követésével,
- kiszámolás bemutatása lengyel formájú kifejezésen.

Verem és Sor

- olyan algoritmus (struktogram) készítése, melyben a feldolgozáshoz csak egyszerű változók, sor, ill. verem használható.

Láncolt ábrázolás, listák feldolgozása

- egyszerű (S1L), egyirányú fejelemes (H1L) listán adott feladat végrehajtása (struktogram készítés),
- kétirányú, fejelemes ciklikus listán (C2L) adott feladat végrehajtása (struktogram készítés). Használhatók a tanult (precede, follow, out) listakezelő algoritmusok, de a feladat kérheti a használt algoritmus megadását is!
- két rendezett lista összefésülése. A listák típusa lehet S1L, H1L, C2L.

Minta feladatok**1. feladat (8p)**

A tanult algoritmust bemutatva készítsük el az alábbi kifejezés **lengyel formáját!**

- Ábrázolja követhető módon a **verem** tartalmát!
- Adja meg a kifejezésben szereplő operátorok **rangsorát!**
- Többszörös hatványozást **jobbról balra** kell végrehajtani!

$$x = -x^2 + 5 * k / (y - z * 3 + s) ^ x^2 - b * d - w$$

2. feladat (12p)

Adott a következő, egész számokat tartalmazó tömb: $\langle 2, 5, 9, 3, 1, 6, 8, 5, 4 \rangle$

Növekvő sorrendbe kell rendezni. Mutassa be, hogyan működne a tanult **mergesort** algoritmus a tömbön! Ábrázolja a szétvágásokat, majd az összefuttatásokat. Az összefuttatások mellé sorolja fel a kulcs-összehasonlításokat! Összesen hány kulcs-összehasonlítás történt? Hány kulcs-összehasonlítás történt volna, ha a maximum kiválasztó rendezést használjuk?

3. feladat (12p)

Egy rádióadónak az a feladata, hogy újra és újra leadja ugyanazt a jelsorozatot. A jelsorozat végén egy szünet-jelet ad, a példákban ezt # jellel jelöltük. Az adást egy szekvenciális fájlban rögzítettük, készítsen algoritmust, mely sor segítségével eldönti, hogy volt-e hiba az adás során. Az input fájl csak egyszer olvasható végig.

Példák:

a a 0 b # a a 0 b # a a 0 b	hibátlan
c 0 b # c 0 b # 0 b	hibás
a a a b c	hibátlan (nem volt ismétlés)
a b c d # a b c d # a b c d a	hibás

4. feladat (12p)

Egy szigorúan monoton növekvően rendezett **H1L** (egyirányú, fejelemes) listával **zsák** adatszerkezetet ábrázoltunk. A zsák abban tér el a halmaztól, hogy egy adott értékből több példány is lehet a zsákban. A lista egy eleme a következő négyesből áll: (**key**, **mult**, **next**). A lista **fejelemére az L** pointer mutat. Készítsen algoritmust, mely a zsák **minden eleméből egyet kivesz** (multiplicitásukat eggyel csökkenti). Ha a multiplicitás 0-ra csökken, az elemet kifűzi a láncból. A kifűzött elemeket egy T pointerű, egyszerű listába (S1L) fűzi. T lista is növekvően rendezett legyen! L lista egyszer járható be, T-be való befűzés $\Theta(1)$ lehet.

5. feladat (16p)

Adott két kétirányú fejelemes ciklikus (**C2L**) lista: **L1** és **L2**. A listák szigorúan monoton növekedően rendezettek, halmazt ábrázolnak. Készítsen `unionIntersection(L1,L2)` néven algoritmust, mely L2 lista megfelelő elemeinek átfűzésével előállítja **L1-ben** a két halmaz **unióját**, míg **L2-ben** keletkezzen a két halmaz **metszete**. Mindkét lista maradjon szigorúan monoton növekedő, műveletigény $O(m+n)$.

FONTOS megjegyzések:

- A struktogramok fejlécének az elkészítése is a feladat része! Adjunk nevet az algoritmusoknak, paramétereiket tüntessük fel, ügyeljünk a helyes paraméter átadási mód (cím-, vagy értékszerinti) kiválasztására! Függvények esetén a visszatérési érték típusát is meg kell adni!
- C2L listák esetén használható az alapl műveletekhez az előadáson bevezetett `follow`, `precede`, és `out`. Ekkor viszont a használt művelet struktogramját is meg kell adni.
- A 4. és 5. feladathoz nem használható az elemek számával arányos méretű tároló (tömb, sor, verem)! A feladatokat az eredeti listaelemek átláncolásával kell megoldani!