

Információ- és kódelmélet

2018/19/2

Otthoni feladatok

Az itt található beadandó feladatok megoldása önálló munka, beadásuk nem kötelező, a megoldás minőségétől függően kisZH, ZH és vizsga plusz jegyek kaphatóak értük.

A beadás határideje a vizsganap (amire a hallgató a Neptunon feljelentkezett) előtti nap 12:00.

A feladat beadása az írásbeli vizsga megírása alól nem mentesít, azt legalább elégséges jegyre (minimum 40 pont) kell megírni. A beküldött feladat bemutatására és értékelésére a szóbeli vizsgán kerül sor.

A megoldások értékelése az alábbi skálával történik:

- 5-15 pont : 1 kisZH jegy
- 15-20 pont : 2 kisZH jegy
- 20-25 pont : 3 kisZH jegy
- 25-30 pont : 4 kisZH jegy
- 30-40 pont : 5 kisZH jegy
- 40-70 : ZH jegy emelés
- 70 felett : Vizsgajegy emelés
- Legjobb megoldás (70 pont felett): Vizsga jeles (ez kerül a Neptunba) + csoki

Minden feladat beadása során be kell adni:

1. A fordítható és futtatható állományt.
2. A fordítás-futtatás és használat lépéseinek listászerű leírását
 - a. Ha ezen a listán végighaladva nem kapjuk meg az elvárt eredményt, a feladatra nem tudunk pontot adni
3. A forráskódot és annak dokumentációját
 - a. A forráskód szerkesztése legyen jó, a függvények, változók és objektumok névválasztása tükrözze azok funkcióját
 - b. A függvények, objektumok legyenek minél rövidebbek, ne legyenek bennük keverve az absztrakciós szintek és felelősségi körök (lásd: Tiszta kód)
 - c. A dokumentációnak nem kell hosszúnak lennie.
4. A feladatokban kért grafikonokat, következtetések szövegesen.

A feladatok több részfeladatra vannak osztva, nem kötelező mindegyiket megoldani. A megoldott részfeladatok több feladtból is kikerülhetnek (de mindegyiknek legyen külön dokumentációja, ami a fenti pontokat tartalmazza).

A beadás emailen (olaha@itk.ppke.hu és lorincz.mate@itk.ppke.hu) egy link megadásával történik, ahonnan a fenti anyagok letölthetőek.

Választható feladatok:

I. Hibajavító kódolás (100 pont)

Írjon szimulációt, mely szimulálja a csatornaátvitelt és valamilyen hibajavító kódolást. (Pl.: mint ez: http://users.itk.ppke.hu/~lorma/select_dmin/)

A szimuláció menete, hogy egy véletlenszerűen generált **u** vektort kódoljon a program, adjon hozzá egy véletlenszerűen (állítható eloszlással) generált hibavektort, majd próbálja meg visszaalakítani.

Az egyes lépések láthatóak és elkülöníthetőek legyenek a program futása során! (Jelenjen meg **u**, **c**, **e**, **v**, **e'**, **u'**)

A kész szimulációval teljesítőképesség analízist is kell készíteni (= grafikonok pl. csatorna **P_b** hibavalószínűségének függvényében a helyesen átvitt üzenetek száma), és azokat szövegesen is ki kell értékelni.

- Pontozás:

- Használható GUI : **15 pont**
- Ismétléses kód implementálása: **5 pont**
- Hamming kód implementálása (paraméterben kérje be **n**-t, generáljon hozzá **k**-t, írja is ki őket)
 - Minden 1 hiba javítása: **5 pont**
 - Minden 2 hiba javítása: **5 pont**
- Golay kód (ciklikus kód) : **15 pont**
 - Ez előadáson és gyakorlaton nem került elő, az utánaolvasás és egy legalább 2 diás összefoglaló írása is része a feladatnak.
- RS kód GF(q) felett **25 pont**
 - A generátort nem kell generálni programból, azok előre legenerálhatóak. Ha nem a programból generálódik, legyen legalább három beépítve, melyek képesek javítani minden 2, 3 és 4 hibát.
 - A feladat az ETA implementálását is tartalmazza
- RS kód GF(2ⁿ) felett **30 pont**
 - A generátort nem kell generálni programból, azok előre legenerálhatóak. Ha nem a programból generálódik, legyen legalább három beépítve, melyek képesek javítani minden 2, 3 és 4 hibát.
 - A feladat az ETA implementálását is tartalmazza

II. Adattömörítés (100)

A feladat egy tömörítő és kicsomagoló program írása, amely képes tetszőleges bináris fájlt betömöríteni és kicsomagolni.

A tömörítés kimenete két fájl, egyik a tömörített adatot tartalmazza, a másik a kicsomagoláshoz szükséges metainformációkat (pl.: szimbólumtábla). Egy valódi tömörítőprogramnál (pl. WinZip) ezek egy fájlba kerülnek, de most fontos, hogy látszódjon, mekkora a tömörített adat és mennyi információ kell a visszaállításához.

Mindegyik algoritmus esetén a feladat része a kiosztott 5 fájl tömörítése (véletlen, egyenletes eloszlású bináris fájl; véletlen, nem egyenletes eloszlású bináris fájl; Karinthy összes verse plain text formátumban, Karinthy összes verse docx formátumban, Karinthy összes verse plain text formátumban GZ-vel tömörítve) és a tapasztalatok rövid leírása. (Melyik fájlt mennyire lehetett tömöríteni, miért?)

A választható tömörítő algoritmusok:

- 1) SF (15 pont)
- 2) Huffman (15 pont)
- 3) Adaptív Huffman (25 pont)
- 4) SFE (15 pont)
- 5) Lempel-Ziv (15-15 pont)

III. Titkosítás (50)

A tananyag harmadik rész a titkosítás, melyből két algoritmust lehet implementálni:

- OTP : 10 pont
- RSA : 40 pont

Mindkét algoritmus bemenete egy tetszőleges fájl és a (publikus) kulcs, kimenete pedig a titkosított fájl.

A feladathoz hozzátartozik egy visszafejtő program is, melyek bemenete a (privát) kulcs, kimenete pedig az eredeti állomány.