

| | | |
|-----------|---------------------------------------|------------|
| MI | Név, felvételi azonosító, Neptun-kód: | pont(90) : |
|-----------|---------------------------------------|------------|

| | | |
|---|---|--|
| Csak felvételi vizsga: <input type="checkbox"/> | csak záróvizsga: <input type="checkbox"/> | közös vizsga: <input type="checkbox"/> |
|---|---|--|

Közös alapképzéses záróvizsga – mesterképzés felvételi vizsga

Mérnök informatikus szak

BME Villamosmérnöki és Informatikai Kar

2011. május 31.

A dolgozat minden lapjára, a kerettel jelölt részre írja fel nevét, valamint felvételi azonosítóját, záróvizsga esetén Neptun-kódját!

A fenti táblázat megfelelő kockájában jelölje X-szel, hogy csak felvételi vizsgát, csak záróvizsgát, vagy közös felvételi és záróvizsgát kíván tenni!

A feladatok megoldásához csak papír, írószer, zsebszámológép használata megengedett, egyéb segédeszköz és a kommunikáció tiltott. A megoldásra fordítható idő: 120 perc. A feladatok után azok pontszámát is feltüntettük.

A megoldásokat a feladatlpra írja rá, illetve ott jelölje. Teszt jellegű kérdések esetén elegendő a kiválasztott válasz betűjelének bekarikázása. Kiegészítendő kérdések esetén, kérjük, adjon világos, egyértelmű választ. Ha egy válaszon javítani kíván, teszt jellegű kérdések esetén írja le az új betűjelet, egyébként javítása legyen egyértelmű.

A feladatlpra írt információk közül csak az eredményeket vesszük figyelembe. Az áttekinthetetlen válaszokat nem értékeljük.

A vizsga végeztével mindenképpen be kell adnia dolgozatát. Kérjük, hogy a dolgozathoz más lapokat ne mellékeljen.

Felhívjuk figyelmét, hogy illegális segédeszköz felhasználása esetén a felügyelő kollegák a vizsgából kizárják, ennek következtében felvételi vizsgája, illetve záróvizsgája sikertelen lesz, amelynek letételét csak a következő felvételi, illetve záróvizsga-időszakban kísérelheti meg újból.

Szakirányválasztás

(Csak felvételi vizsga esetén kell kitölteni)

Kérem, az alábbi táblázatban jelölje meg, mely szakirányon kívánja tanulmányait folytatni. A táblázatban a szakirány neve mellett számmal jelölje a sorrendet: 1-es szám az első helyen kiválasztott szakirányhoz, 2-es a második helyen kiválasztotthoz tartozik stb. Nem kell az összes szakirány mellé számot írni, de legalább egy szakirányt jelöljön meg. Egy sorszám csak egyszer szerepeljen.

| szakirány neve | gondozó tanszék | sorrend |
|--|-----------------|---------|
| Alkalmazott informatika szakirány | AAIT | |
| Autonóm irányító rendszerek és robotok szakirány | IIT | |
| Hálózatok és szolgáltatások szakirány | TMIT | |
| Hírközlő rendszerek biztonsága szakirány | HIT | |
| Intelligens rendszerek szakirány | MIT | |
| Médiainformatika szakirány | TMIT | |
| Rendszerfejlesztés szakirány | IIT | |
| Számításelemélet szakirány | SZIT | |
| Szolgáltatásbiztos rendszertervezés szakirány | MIT | |

| | | |
|-----------|---------------------------------------|---|
| AL | Név, felvételi azonosító, Neptun-kód: | pont(30) : <input style="width: 100px;" type="text"/> |
| | | |

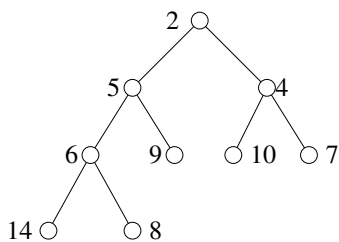
1. Legyen $f(n) = 3f(n - 2) + 5n + 8$. Igaz-e, hogy

(i) $f = O(n^2)$?

(ii) $f = O(4^n)$?

pont(2):

2. Az alábbi kupacon hajtsa végre a BESZÚR(3) műveletet!

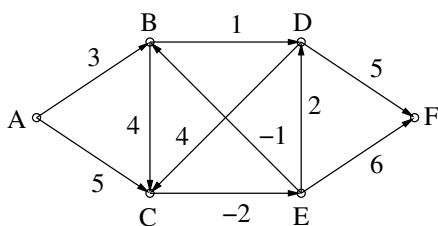


pont(2):

3. Az $1, 2, \dots, 20$ számokból 7 elemű részhalmazokat képezünk. Hány olyan van közöttük, amelyek metszi az $\{1, 2\}$ halmazt? (A pontos szám nem fontos, elég egy zárt formulával megadni.)

pont(2):

4. Az alábbi gráfon a Bellman-Ford algoritmust kezdtük el alkalmazni. Fejezze be az algoritmust! Mit adnak meg az utolsó sorbeli számok az algoritmus végén?



| | A | B | C | D | E | F |
|----|---|---|---|----------|----------|----------|
| 1. | 0 | 3 | 5 | ∞ | ∞ | ∞ |
| 2. | 0 | 3 | 5 | 4 | 3 | ∞ |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

pont(4):

5. Adott n pozitív szám x_1, x_2, \dots, x_n , valamint egy k pozitív egész. Azt kell eldöntenünk, hogy vannak-e olyan $I_1, I_2, \dots, I_k \subseteq \{1, 2, \dots, n\}$ diszjunkt indexhalmazok ($I_j \cap I_t = \emptyset$, ha $j \neq t$), melyekre $\cup_{j=1}^k I_j = \{1, 2, \dots, n\}$ és minden $1 \leq j \leq k$ esetén $\sum_{i \in I_j} x_i \leq 1$.

Melyik ismert feladatot írja le a kérdés?

pont(4):

6. Tegyük fel, hogy P tartalmazza az NP osztályt. Tekintsük a következő \mathcal{A} és \mathcal{B} problémát.

\mathcal{A} : Adott egy G irányítatlan gráf.

Kérdés, hogy van-e G -ben legalább 5 pontú teljes részgráf.

\mathcal{B} : Adott egy G irányítatlan gráf és egy $k > 0$ egész szám.

Kérdés, hogy van-e a G gráfban pontosan k pontú teljes részgráf.

Van-e \mathcal{A} -ról \mathcal{B} -re polinomiális visszavezetés (Karp-redukció)? Válaszát röviden indokolja is!

pont(4):

7. Az A tömb, amely n csupa különböző számot tartalmaz. Adjon algoritmust, amely az A tömb ismeretében $O(n^2)$ lépésben meghatározza az A -beli leghosszabb monoton csökkenő részsorozat hosszát! (Azaz a maximális olyan k számot, melyre található olyan $i_1 < i_2 < \dots < i_k$, hogy $A[i_1] > A[i_2] > \dots > A[i_k]$.)

pont(6):

8. Két marslakó a következő játékot játssza. Az egyik marslakó mond két 10 marsi karakterből álló szót (természetesen marsi nyelven), A -t és B -t. A másik marslakó feladata tetszőleges számú lépésben eljutni az A szótól a B szóig úgy, hogy minden lépésben egyetlen karaktert módosíthat, azonban ezt csak úgy teheti, hogy a közbülső lépésben kapott szó szintén értelmes marsi szó legyen. (Az eredetileg kapott A és B szavak értelmes szavak.) Adjon algoritmust, amely az A és B szavak, valamint az összes 10 karakterből álló értelmes marsi szó L listája alapján meghatározza, hogy megoldható-e a marslakó feladata! Az algoritmus lépésszáma legyen $O(|L|^2)$, ahol $|L|$ az L lista hosszát jelöli! (A marsi karakterek számáról csak annyit tudunk, amennyi a feladat szövegéből következik.)

pont(6):

| | | |
|----------|---------------------------------------|------------|
| H | Név, felvételi azonosító, Neptun-kód: | pont(15) : |
|----------|---------------------------------------|------------|

1. Mi annak a protokollnak a neve, amelynek segítségével az IP-cím ismeretében meg lehet határozni az adatkapcsolati rétegbeli címet?

pont(2):

2. Mit csinál egy IPv4 router, ha akkora töredékekben érkezik hozzá egy csomag, amelyek kicsit nagyobbak, mint a kimenő porton használt adatkapcsolati keret payloadjának mérete?

- a) Megnöveli az adatkapcsolati réteg payloadjának méretét.
- b) Összerakja a töredékeket az eredeti csomaggá, és újratördeli a megfelelő méretre.
- c) Eldobja a csomagot, mert töredéket nem szabad tovább tördelni.
- d) A többi válasz közül egyik sem helyes.

pont(2):

3. Milyen információt juttatnak el a csomópontok és kiknek a link-state (összekötés-állapot) routing módszer esetén?

- a) A csomópontok elmondják a hálózatról alkotott elképzeléseiket mindenkinek.
- b) A csomópontok elmondják a szomszédaiknak a velük kapcsolatos tapasztalataikat.
- c) A csomópontok elmondják a hálózatról alkotott elképzeléseiket szomszédaiknak.
- d) A többi válasz közül egyik sem helyes.

pont(2):

4. Az alábbiak közül mely állítás(ok) igaz(ak) az Ethernet backoff stratégiájára?

- a) Tisztán exponenciális.
- b) Tisztán lineáris.
- c) Lehetővé teszi az adaptációt a felhasználók számára.
- d) A 3. ütközés után a 0, ..., 7 intervallum lesz a sorsolási intervallum.
- e) A backoff értékét résidőkben számoljuk.
- f) A többi válasz közül egyik sem helyes.

pont(2):

5. Az alábbiak közül mely paraméter(ek) szükséges(ek) egy alkalmazás megcímzésre az IP-hálózaton keresztül?

- a) IP-verzió és IP-cím.
- b) Szállítási rétegbeli protokoll azonosítója.
- c) Alkalmazás futtatható állományának fájlneve.
- d) Az alkalmazás memóriacíme.
- e) A hálózati csatoló fizikai címe.
- f) Szállítási rétegbeli protokoll alkalmazáshoz rendelt portszáma.

pont(2):

6. Nevezze meg (magyarul vagy angolul) azt a jellemzően többportos eszközt, amely akár több különféle hálózat között is átjárást biztosíthat újraakerevezéssel anélkül, hogy a hordozott hálózati rétegbeli csomagot értelmezné, feldolgozná!

pont(2):

7. Az *A* és *B* végpont közötti kommunikáció során *A* végpont utolsóként elküldött TCP PDU-jában a sorszám (sequence number) 4740, a hasznos adatrész 150 byte. *B* válaszként küldött TCP PDU-jában az ACK-szám 4350. Hány byte-nyi adatot küldhet még *A* a következő nyugta megérkezéséig, ha az ablakméret 600?

pont(3):

| | | |
|----------|---------------------------------------|------------|
| O | Név, felvételi azonosító, Neptun-kód: | pont(15) : |
|----------|---------------------------------------|------------|

1. Az alábbi állítások közül melyik *hamis* az operációs rendszerek tipikus belső felépítésével kapcsolatban?

- a) Az operációs rendszer magja (kernel) csak az alapfunkciókat, pl. memóriakezelés, folyamat- és szálkezelés, CPU ütemezés tartalmazza.
- b) A felhasználói programok nem érhetik el direkt módon (pl. I/O gépi utasítások) a hardver elemeket.
- c) Az alkalmazói programok függvényhívásokkal vagy szubrutinhívásokkal érik el az operációs rendszer szolgáltatásait.
- d) Az operációs rendszerekben mindig találunk egy alsó, hardverközeli réteget, amely elfedi a hardware elemek specialitásait, és absztrakt hozzáférést tesz lehetővé a hardverhez.

pont(2):

2. Az alábbi esetek közül melyik *nem* hozza működésbe az operációs rendszert, ha a számítógép éppen egy felhasználói programot futtat?

- a) A futó program a yield() rendszerhívás meghívásával lemond a futás jogáról.
- b) A hálózati interfészen beérkezik egy IP csomag, amely hardvermegszakítást (HW interrupt) okoz.
- c) A felhasználói program egy a fizikai memóriában is megtalálható virtuális memóriára ír.
- d) A felhasználói program user módban illegális gépi utasítást kísérel meg végrehajtani, aminek hatására a CPU kivételt (exception) hajt végre.

pont(2):

3. Melyik állítás *igaz* minden esetben a folyamatokra (process)?

- a) A folyamat szekvenciális program.
- b) A folyamat végrehajtás alatt álló program.
- c) A folyamatok közötti kommunikáció közös memórián keresztül történik.
- d) A folyamatok létrehozása és megszüntetése kevésbé erőforrás igényes a szálakkal összehasonlítva.

pont(2):

4. A folyamatok egyszerű állapotátmeneti diagramja alapján mely állítás *hamis* a következő állításokból kooperatív (nem preemptív) operációs rendszer esetén?

- a) A folyamatok „Futásra kész” állapotba kerülnek létrehozásuk után.
- b) Az I/O löket alatt a folyamatok „Várakozó” állapotban várnak a rendszerhívás befejezésére.
- c) A processzort a futó folyamatától az operációs rendszer elveheti.
- d) A folyamat csak futó állapotból fejeződhet be.

pont(2):

5. Mely processzorütemezési algoritmusokkal kapcsolatos állítás *igaz* az alábbiak közül?

- a) A körforgó (RR: Round Robin) algoritmus a legrövidebb löketidejű (SJF: Shortest Job First) algoritmus preemptív változata.
- b) A legrégebben várakozó (FCFS: First Come First Serve) algoritmus átmegy a körforgó (RR: Round Robin) algoritmusba, ha túl hosszú időszületet választunk.
- c) A körforgó (RR: Round Robin) ütemező algoritmusban megjelenhet a konvoj hatás.
- d) A legrövidebb hátralévő löketidejű (SRTF, Shortest Remaining Time First) algoritmus prioritásos algoritmus.

pont(2):

6. Melyik állítás *hamis* a virtuális tárkezelést használó rendszerekkel kapcsolatban?

- a) Az előretekintő lapozás (anticipatory paging) mindig növeli a virtuális tárkezelés teljesítményét.
- b) Virtuális memóriakezelés esetén a rendelkezésre álló központi (fizikai) memóriánál nagyobb fizikai memória igényű programok is futtathatók.
- c) A futó programok memóriájának csak a ténylegesen használt része kell, hogy megtalálható legyen a központi (fizikai) memóriában.
- d) A virtuális tárkezelés a felhasználói programokat fejlesztők számára láthatatlan, azzal nem kell törődni, csupán a program tényleges futási sebességét fogja befolyásolni.

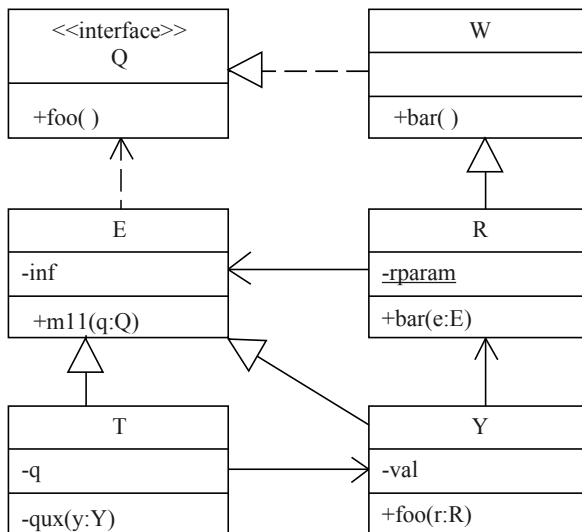
pont(2):

7. Rajzolja fel, hogyan történik a címtranszformáció lapszervezés esetén egyszintű laptábla alkalmazásával! Az asszociatív gyorsítótár felrajzolása nem szükséges.

pont(3):

| | | |
|-----------|---------------------------------------|------------|
| S1 | Név, felvételi azonosító, Neptun-kód: | pont(10) : |
|-----------|---------------------------------------|------------|

1. Az alábbi UML2 diagram alapján – a kulcs felhasználásával – jellemezze az állításokat!



- A – mindkét tagmondat igaz és a következtetés is helyes (+ + +)
- B – mindkét tagmondat igaz, de a következtetés hamis (+ + -)
- C – csak az első tagmondat igaz (+ -)
- D – csak a második tagmondat igaz (- +)
- E – egyik tagmondat sem igaz (- -)

(i) E helyettesíthető R-rel, mert R interfésze kompatibilis E interfészével.

(ii) T-nek m11(q:Q) metódusa nem kaphat paraméterként R-et, mert T nem függ R-től.

pont(2):

2. A szoftverfejlesztés vízésmodellje szerint a fejlesztésnek melyik az első fázisa?

pont(2):

3. Nevezzen meg egy statikus és egy dinamikus verifikációs technikát!

Statikus:

Dinamikus:

pont(2):

4. Pali készít egy tákolmányt, majd azt berakja a vitrinbe. Később Feri elkéri Palitól a tákolmányt, és azonnal ráírja a dátumot. Rajzoljon UML2 szekvenciadiagramot!

pont(2):

5. Elkészítjük az alábbi `O` osztály két példányát, `o1`-et és `o2`-t. Ezt követően – sorrendben – végrehajtjuk a következő műveleteket:

`o2.x = 3; o1.x = -2;`
`o1.y = o2.x + 5;`
`o2.y = o2.x + o1.y;`

| |
|--|
| O |
| <code>int x = 11</code> <code>int y = -4</code> |
| <code>private xx(): int</code> |

Mennyi lesz az `o2.y` változó értéke?

`o2.y =`

pont(2):

| | | |
|-----------|---------------------------------------|------------|
| S2 | Név, felvételi azonosító, Neptun-kód: | pont(10) : |
|-----------|---------------------------------------|------------|

1. Egy-két mondatban adja meg, milyen általános problémát old meg a Composite (Összetett) tervezési minta!

pont(2):

-
2. Milyen általános problémát old meg a Factory Method (Metódusgyár) tervezési minta?

pont(2):

-
3. Rajzolja fel általánosságában vagy egy példára vonatkozóan a Factory Method (Metódusgyár) minta osztály-diagramját!

pont(2):

4. Az előző feladat osztálydiagramjára építve ismertesse általánosságában vagy egy példa alapján a Factory Method minta működését, jellemezze a benne szereplő osztályokat!

pont(2):

5. Hasonlítsa össze a kliens és a kiszolgáló oldali szkript szerepét a webalkalmazásokra vonatkozóan!

pont(2):

| | | |
|-----------|---------------------------------------|------------|
| AD | Név, felvételi azonosító, Neptun-kód: | pont(10) : |
|-----------|---------------------------------------|------------|

1. Egy adatbázis rekordjainak a kulcsértékük szerinti elérését „vödörös hash” szervezéssel szeretnénk gyorsítani. 10 millió rekord található az adatbázisban, melyek hossza fix 240 byte, benne a kulcs 25 byte. A háttértár blokkelérésű, egy blokk kapacitása (a fejrészt nem számítva) 4000 byte. Minden mutató 8 byte-on tárolt. Az alkalmazás nem teszi lehetővé, hogy 5 blokkelérésnél több idő legyen a keresésre. Mennyi legyen a vödörök minimális száma és mekkora lesz ekkor a hash-tábla mérete? (Tételezze fel, hogy a vödörkatalógus kereséskor memóriában tartható és a hash függvény egyenletesen osztja el a kulcsokat.)

pont(2):

2. Határozza meg az atomi attribútumokat tartalmazó $R(L, M, N, O)$ relációs séma normálformáját az

$$F = \{M \rightarrow O, LM \rightarrow LN, N \rightarrow M, NO \rightarrow M\}$$

függéshalmaz mellett!

pont(2):

3. Ekvivalens-e a következő két függéshalmaz?

$$\{AB \rightarrow C, AC \rightarrow B, A \rightarrow BC\} \quad \text{és} \quad \{A \rightarrow B, A \rightarrow C, AC \rightarrow BC\}$$

pont(2):

4. Mutasson példát olyan relációs sémára, amely nem bontható fel veszteségmentesen és függőségörzően BCNF részsémákra!

pont(2):

5. Érveljen a következő okfejtés mellett vagy ellen:

Minden BCNF séma egyben 3NF is. Mivel minden 3NF sémára illeszkedő reláció tartalmazhat redundanciát funkcionális függés következtében, ezért a BCNF sémára illeszkedő reláció is tartalmazhat redundanciát funkcionális függés következtében.

pont(2):