

| | | | |
|----------|------|-------------|-----------|
| A | Név: | Neptun kód: | Pontszám: |
|----------|------|-------------|-----------|

1. Feladat (8 pont)

Lásd be, hogy a Θ függvényosztály zárt a számmal történő szorzásra, azaz, ha $f = \Theta(g)$ és $c > 0$, akkor $cf = \Theta(g)$.

2. Feladat (10 pont)

Hozd postfix (fordított lengyelforma) alakra a következő kifejezést, majd értékeld ki az $a = 0, b = 2, c = 4$ értékekkel. Add meg az átalakítás, illetve a kiértékelés lépéseit oly módon, hogy a bemenet minden elemének feldolgozásakor írd le az aktuális veremtartalmat, valamint a kiírandó értéket (változatlan verem esetén nem szükséges ismételni).

$$x := a/3 + (5 \cdot c - 4^b)/(c - 2)^b$$

3. Feladat (14 pont)

Valósítsuk meg a *CorrectParenthesis(in)* algoritmust a helyes zárójelezés azon problémájának eldöntésére, amikor '(', ')', valamint '[,]' zárójelpárok szerepelhetnek a sorozatban, és mindkét zárójel tartalmazhatja a másikat. Pl. '()', '(())', '([])', '[() ([])]' helyes zárójelezések, '(()', '()', '([])' nem azok. A megvalósításhoz használj verem adatszerkezetet, a beolvasáshoz a *Read(< bemenet >, < karakter >)* függvényt. A bemenet végét ϵ jelzi (a hossza nem ismert). A bemeneten csak zárójelek szerepelnek.

4. Feladat (18 pont)

Valósítsd meg a *ReverseStrings(in, out)* algoritmust, amely egy bemenő sorozatból, szóközzel elválasztott szavak sorrendjét tudja megfordítani. Pl.:

Péter Anna Gábor \rightarrow *Gábor Anna Péter*.

Feltehető, hogy a szavak között pontosan egy szóköz található. Az algoritmushoz felhasználhatóak a veremműveletek, a beolvasás *Read(< bemenet >, < karakter >)*, valamint a kiírás *Write(< kimenet >, < karakter >)* függvényei. A bemenet végét ϵ jelzi.

Add meg, és indokold az algoritmus műveletigényét is.

5. Feladat (10 pont)

Valósítsd meg a sor *IsEmpty(q)* és *Out(q)* műveleteit aritmetikai reprezentáció mellett. Add meg az algoritmusokat, valamint a műveletigényeket, illetve ábra segítségével magyarázd el a működési elvet.

Jó munkát!

Giachetta Roberto

| | | | |
|----------|------|-------------|-----------|
| B | Név: | Neptun kód: | Pontszám: |
|----------|------|-------------|-----------|

1. Feladat (8 pont)

Lásd be, hogy a Θ függvényosztály zárt az összeadásra, azaz, ha $f = \Theta(h)$ és $g = \Theta(h)$, akkor $f + g = \Theta(h)$.

2. Feladat (10 pont)

Hozd postfix (fordított lengyelforma) alakra a következő kifejezést, majd értékeld ki az $a = 0, b = 2, c = 1$ értékekkel. Add meg az átalakítás, illetve a kiértékelés lépéseit oly módon, hogy a bemenet minden elemének feldolgozásakor írd le az aktuális veremtartalmat, valamint a kiírandó értéket (változatlan verem esetén nem szükséges ismételni).

$$x := (2 - a)^{(3 - c)}/4 + a/(b + 2) \cdot 3$$

3. Feladat (10 pont)

Valósítsd meg a verem $Top(s)$ és $Pop(s)$ műveleteit aritmetikai reprezentáció mellett. Add meg az algoritmusokat, valamint a műveletigényeket, illetve ábra segítségével magyarázd el a működési elvet.

4. Feladat (14 pont)

Valósítsd meg az $IsRepeating(in, w)$ algoritmust, amely eldönti, hogy egy szekvenciális bemenő sorozat w^k alakú-e, azaz a paraméterben megadott szövegrész ismétlődik benne (legalább egyszer). A megvalósításhoz használj sor adatszerkezetet, a beolvasáshoz a $Read(< bemenet >, < karakter >)$ függvényt. A bemenet végét $< bemenet > = \varepsilon$ jelzi (a hossza nem ismert).

Az algoritmus visszatérési értékben jelezze a bemenet helyességét.

5. Feladat (18 pont)

Egy sor működése szimulálható két verem segítségével, kihasználva a vermeknek azon képességét, hogy az elemek sorrendjét meg tudják fordítani. Add meg ilyen reprezentáció mellett a sor $First(q)$ és $In(q, d)$ műveleteit, ahol tehát $q = (s_1, s_2)$. Feltételezhető, hogy a vermek reprezentációja láncolt.

Add meg, és indokold az algoritmusok műveletigényét is.

Jó munkát!

Giachetta Roberto

| | | | |
|----------|------|-------------|-----------|
| C | Név: | Neptun kód: | Pontszám: |
|----------|------|-------------|-----------|

1. Feladat (8 pont)

Lásd be, hogy a Θ függvényosztályra teljesül a tranzitivitás, azaz $f = \Theta(g)$, illetve $g = \Theta(h)$ esetén $f = \Theta(h)$.

2. Feladat (10 pont)

Hozd postfix (fordított lengyelform) alakra a következő kifejezést, majd értékeld ki az $u = 100, v = 10, w = 1$ értékekkel. Add meg az átalakítás, illetve a kiértékelés lépéseit oly módon, hogy a bemenet minden elemének feldolgozásakor írd le az aktuális veremtartalmat, valamint a kiírandó értéket (változatlan verem esetén nem szükséges ismételni).

$$x := u - 4 \cdot (w + 4)^2 + (2 - v)/(w + 3)$$

3. Feladat (14 pont)

Valósítsd meg az *IsMirrorWord(in)* algoritmust, amely eldönti, hogy egy szekvenciális bemenő sorozat tükörszó-e, azaz az eleje tükröződik a végében. Tükörszónak csak a páros hosszú szavakat tekintjük (pl. arra, inni, ebbe) Az üres bemenet is tükörszó. A megvalósításhoz használj verem adatszerkezetet, a beolvasáshoz a *Read(< bemenet >, < karakter >)* függvényt. A bemenet végét ϵ jelzi (a hossza nem ismert). Az algoritmus visszatérési értékben jelezze a bemenet helyességét.

4. Feladat (10 pont)

Valósítsd meg a sor *In(q, d)* és *First(q)* műveleteit láncolt reprezentáció mellett. Add meg az algoritmusokat, valamint a műveletigényeket, illetve ábra segítségével magyarázd el a működési elvet.

5. Feladat (18 pont)

Egy verem működése szimulálható egy sor segítségével, köszönhetően annak, hogy a sor tartalma körkörösen változtatható elemek kivételével, majd újra behelyezésével. Add meg ilyen reprezentáció mellett a verem *Push(s, d)* és *Top(s)* műveleteit, ahol tehát $s = (q)$. Feltételezhető, hogy a sor reprezentációja láncolt.

Add meg, és indokold az algoritmusok műveletigényét is.

Jó munkát!

Giachetta Roberto

| | | | |
|----------|------|-------------|-----------|
| D | Név: | Neptun kód: | Pontszám: |
|----------|------|-------------|-----------|

1. Feladat (8 pont)

Lásd be a O és Ω függvényosztályokra teljesülő felcserélt szimmetriát, azaz $f = O(g)$ esetén $g = \Omega(f)$.

2. Feladat (10 pont)

Hozd postfix (fordított lengyelforma) alakra a következő kifejezést, majd értékeld ki az $u = 1, v = 2, w = 2$ értékekkel. Add meg az átalakítás, illetve a kiértékelés lépéseit oly módon, hogy a bemenet minden elemének feldolgozásakor írd le az aktuális veremtartalmat, valamint a kiírandó értéket (változatlan verem esetén nem szükséges ismételni).

$$x := (7 - u)/(2^v + 2) \cdot 5^v - (w + 3)$$

3. Feladat (10 pont)

Valósítsd meg a verem $Push(s, d)$ és $IsEmpty(s)$ műveleteit aritmetikai reprezentáció mellett. Add meg az algoritmusokat, valamint a műveletigényeket, illetve ábra segítségével magyarázd el a működési elvet.

4. Feladat (14 pont)

Valósítsd meg a postfix alak (fordított lengyelforma) kiértékelés $EvalRPN(in)$ algoritmusát. A kifejezés a bemenetről a $Read(< bemenet >, < szimbólum >)$ művelettel olvasható, amely egy operandust, vagy operátort olvas be (más nincs a bemeneten, illetve az operátorok mind kétoperandusúak, de a kifejezés lehet hibás is). A beolvasott értéket ellenőrizhető az $Operator(< szimbólum >)$ függvénnyel, amely igazgal tér vissza operátorra. Egy kétoperandusú kifejezés az $Eval(< operátor >, < balérték >, < jobbérték >)$ művelettel értékelhető ki. A megvalósításhoz használj verem adatszerkezetet.

5. Feladat (18 pont)

Valósítsd meg az $IsRepeating(in)$ algoritmust, amely eldönti, hogy egy szekvenciális bemenő sorozat WW alakú-e, azaz eleje ismétlődik a végében. A megvalósításhoz használj sor adatszerkezetet, a beolvasáshoz a $Read(< bemenet >, < karakter >)$ függvényt. A bemenet végét $< bemenet > = \varepsilon$ jelzi (a hossza nem ismert).

Add meg, és indokold az algoritmus műveletigényét is.

Jó munkát!

Giachetta Roberto