

Tehtävä 1: Vastaukset

Kysymys 1.

jono	(a)	(b)
ate	hyväksyy	hyväksyy
aate	hyväksyy	hylkää
atue	hylkää	hylkää
aatutue	hylkää	hyväksyy
aamute	hyväksyy	hylkää
atmut	hylkää	hylkää

Kysymys 2.

(a) 25:59

(c) Miksi seuraava lauseke korjaisi (a)-kohdan ongelmia?

$(([01]?[0-9])|(2[0-3])):[0-5][0-9]$

– Nyt $(([01]?[0-9])|(2[0-3]))$ rajoittaa tunnit välille 0-19 tai 20-23

Kysymys 3.

Jonossa on kolmella jaollinen määrä nollia.

(a) Esimerkiksi

0101011111

0000001111

1111111111

(b) Esimerkiksi (tässä ei voi olla yhtään ykköstä)

00000

00

0

(c) hyväksyy tyhjän jonon, perustelu: kaikki osia voi olla nolla kertaa.

Kysymys 4. Esimerkiksi

(a) (b)

00000 11111

10001 10101

11100 11101

00100 01101

01000 01010

Tietotekniikan kilpailu 23.1.2014

Kysymys 5. Pitää hyväksyä myös pelkkä jono 00 sekä jono joka alkaa osajonolla 00, mutta jonon ei ole pakko alkaa osajonolla 00

Lauseke	Hyväksyy
0*1*	
[01]+	
[01]*00[01]*	X
00[01]*	
[01]+00[01]+	

Kysymys 6.

Jonosta löytyy ensin jostakin kohdasta osajono 00 ja sitten heti tai myöhemmin osajono 11, esimerkiksi

- [01]*00[01]*11[01]*
- (0|1)*00(0|1)*11(0|1)*
- .*00.*11.*

Jonon voi myös tarkistaa kahdella lausekkeella. Jos jono on tarkistettu ensiksi lausekkeella [01]* voidaan olla varmoja, että jonossa on vain merkkejä 0 tai 1, jolloin merkintä .* on sama kuin [01]* tai (0|1)*. Tätä samaa merkintää käytetään myös vastauksessa 7.

Kysymys 7.

Jonossa löytyy joko 00 tai 11, esimerkiksi

- [01]*((00)|(11))[01]*
- (0|1)*((00|11))(0|1)*
- .*((00)|(11)).* kts. kysymys 6

Kysymys 8.

Jonossa ei ole kahta peräkkäistä nollaa (eli jonosta ei löydy 00), esimerkiksi

- ((01)|1)*0?
- 1*(01+)*0?

Kysymys 9.

Jonossa ei ole 00 eikä 11, esimerkiksi

- ((01)*0?)|((10)*1?)
- (0?(10)*1?)
- 1?(01)*0?

Tehtävä 2: Vastaukset

Kysymys 10. Java ja Python

Kysymys 11.

```
/**
 * Tutkii onko merkkijono jossa on 0 ja 1 arvoja
 * hyväksyttävä vai ei.
 * @param jono tutkittava jono
 * @return onko jono hyväksyttävä
 * @example
 * <pre name="test">
 * // Automaatti joka hyväksyy jonot jotka ovat muotoa [01]*00[01]*11[01]*
 * tutkiIfLaskurix00x11x($syote) === $tulos;
 *
 * $syote      | $tulos
 * -----|-----
 * ""          | false
 * "1"         | false
 * "11"        | false
 * "10"        | false
 * "0011"      | true
 * "01011"     | false
 * "010011"    | true
 * "010010110" | true
 * "00101"     | false
 * "01001001"  | false
 * "01001011a" | false
 * </pre>
 */
public static boolean tutkiIfLaskurix00x11x(String jono) {
    int nollia = 0;
    int ykkosia = 0;
    for (int i = 0; i < jono.length(); i++) {
        int arvo = jono.charAt(i) - '0';
        if ( !(arvo == 0 || arvo == 1) ) return false;
        if ( arvo == 0 && nollia < 2 ) nollia++;
        if ( arvo == 0 && nollia == 2 && ykkosia < 2 ) ykkosia=0;
        if ( arvo == 1 && nollia < 2 ) nollia = 0;
        if ( arvo == 1 && nollia == 2 && ykkosia < 2 ) ykkosia++;
    }
    return ( ykkosia == 2);
}
```

```

/**
 * Tutkii onko merkkijono jossa on 0 ja 1 arvoja
 * hyväksyttävä vai ei.
 * @param jono tutkittava jono
 * @return onko jono hyväksyttävä
 * @example
 * <pre name="test">
 * // Automaatti joka hyväksyy jonot jotka ovat muotoa [01]*00[01]*11[01]*
 * tutkiRegExp00x11x($syote) == $tulostulos;
 *
 * $syote      | $tulostulos
 * -----|-----
 * ""          | false
 * "1"         | false
 * "11"        | false
 * "10"        | false
 * "0011"      | true
 * "01011"     | false
 * "010011"    | true
 * "010010110" | true
 * "01001001"  | false
 * "01001011a" | false
 * </pre>
 */
public static boolean tutkiRegExp00x11x(String jono) {
    return Pattern.matches("[01]*00[01]*11[01]*", jono);
}

```

```

/**
 * Tutkii onko merkkijono jossa on 0 ja 1 arvoja
 * hyväksyttävä vai ei.
 * Tutkitaan äärellisellä automaatilla
 *
 *
 * 1 0      0 0 1      1 01
 * 0 ----> 1 --> 2 ----> 3 ----> 4
 * ^      |      ^      |
 * +-----+    +-----+
 *      1          0
 *
 * @param jono tutkittava jono
 * @return onko jono hyväksyttävä
 * @example
 * <pre name="test">
 * // Automaatti joka hyväksyy jonot jotka ovat muotoa [01]*00[01]*11[01]*

```

```
* tutkiIFAutomx00x11x($syote) === $tulostulos;
*
*   $syote      | $tulostulos
*   -----|-----
*   ""          | false
*   "1"         | false
*   "11"        | false
*   "10"        | false
*   "0011"      | true
*   "01011"     | false
*   "010011"    | true
*   "010010110" | true
*   "01001001"  | false
*   "01001011a" | false
* </pre>
*/
public static boolean tutkiIFAutomx00x11x(String jono) {
    int tila = 0;
    for (int i = 0; i < jono.length(); i++) {
        int arvo = jono.charAt(i) - '0';
        if ( !(arvo == 0 || arvo == 1) ) return false;
        if ( tila == 0 && arvo == 0 ) tila = 1;
        else if ( tila == 1 && arvo == 0 ) tila = 2;
        else if ( tila == 1 && arvo == 1 ) tila = 0;
        else if ( tila == 2 && arvo == 1 ) tila = 3;
        else if ( tila == 3 && arvo == 0 ) tila = 2;
        else if ( tila == 3 && arvo == 1 ) tila = 4;
    }
    return tila == 4;
}

public static boolean tutkiAutomaatti2x00x11x(String jono) {
    Automaatti2 automaatti = new Automaatti2("abcdE");
    automaatti.addSiirtymat("ABA,BCA,CCD,DCE");
    automaatti.tutki(jono);
    return automaatti.isHyvaksynta();
}
```