

SimulazioneAlternativaConSoluzioni.java

```
1 public class SimulazioneAlternativaConSoluzioni {
2     /** ESERCIZIO 2.
3      * Scrivere un metodo ricorsivo dicotomico e2 con
4      * le seguenti caratteristiche:
5      * -) e2 ha un parametro formale di tipo array di
6      * interi, e un parametro s di tipo int.
7      * -) e2 restituisce true quando:
8      * a) l'array non e' nullo e
9      * b) almeno uno degli elementi e' > s. */
10    public static boolean e2(int[] x, int s) {
11        boolean risultato = (x != null);
12        if (risultato) {
13            risultato = x.length==0;
14            if(!risultato)
15                risultato = e2(x, 0, x.length, s);
16        }
17        return risultato;
18    }
19
20    public static boolean e2(int[] x, int l, int r, int s) {
21        if (l + 1 == r)
22            return (x[l] > s);
23        else {
24            int m = (l + r) / 2;
25            return e2(x, l, m, s) || e2(x, m, r, s);
26        }
27    }
28
29
30
31
32
33
34
35
36
37
38
39
```

## SimulazioneAlternativaConSoluzioni.java

```
40
41
42 /** ESERCIZIO 3.
43  * Siano dati:
44  * -) il metodo parity, qui sotto definito, da
45  * applicare esclusivamente ad un parametro
46  * attuale con almeno un elemento (a.length>=1)
47  * -) il predicato P(i) seguente:
48  *
49  * "Alla sua uscita, parity(a,i)
50  * rende vero a[i]==(i%2==0)".
51  *
52  * 1) Scrivere il predicato P(0).
53  * 2) Scrivere il predicato P(i-1) ==> P(i).
54  * 3) Dimostrare che P(0) e' vero.
55  * 4) Dimostrare che P(i-1) ==> P(i) e' vero,
56  * ragionando induttivamente. */
57 static void parity(boolean[] a, int i) {
58     if (i < a.length) {
59         if (i == 0)
60             a[i] = true;
61         else {
62             parity(a, i - 1); //(A)
63             a[i] = !a[i - 1]; //(B)
64         }
65     }
66 }
67
68
69
70
71
72
73
74
75
76
77
78
```

## SimulazioneAlternativaConSoluzioni.java

```
79
80
81  /* Soluzione possibile.
82  * 1) P(0) e'
83  *    "Alla sua uscita, parity(a,0)
84  *    rende vero a[0]==(0%2==0)".
85  *
86  * 2) P(i-1) ==> P(i) e'
87  *    "Alla sua uscita, parity(a,i-1)
88  *    rende vero a[i-1]==((i-1)%2==0)"
89  *    ==> "Alla sua uscita, parity(a,i)
90  *    rende vero a[i]==(i%2==0)".
91  *
92  * 3) Osserviamo che:
93  *    -- per definizione, parity(a,0) esegue a[0]=true
94  *    e poi termina.
95  * Quindi:
96  *    a[0] == true == (0%2==0).
97  *
98  * 4) Per ipotesi induttiva assumiamo P(i-1) vero:
99  *
100  *    "Alla sua uscita, parity(a,i-1)
101  *    rende vero a[i-1]==((i-1)%2==0)"
102  *
103  * e' vero.
104  *
105  * --- Caso 1) Assumiamo che i-1 sia pari.
106  * Quindi, per ipotesi, e' vero:
107  *
108  *    a[i-1]==((i-1)%2==0)
109  *
110  * nel punto (A) che e' seguito dalla assegnazione
111  * 'a[i] = !a[i-1]'. Siccome i-1 e' pari, allora i
112  * e' dispari e possiamo scrivere:
113  *
114  *    a[i] == !a[i-1]
115  *    == !((i-1)%2==0)
116  *    == !true
117  *    == false
```

## SimulazioneAlternativaConSoluzioni.java

```
118      *          == (i%2==0).
119      *
120      * vero nel punto (B) che e' esattamente P(i).
121      *
122      * --- Caso 2) Assumiamo che i-1 sia dispari.
123      * Quindi, per ipotesi, e' vero:
124      *
125      *      a[i-1]==((i-1)%2==0)
126      *
127      * nel punto (A) che e' seguito dalla assegnazione
128      * 'a[i] = !a[i-1]'. Siccome i-1 e' dispari, allora
129      * i e' pari e possiamo scrivere:
130      *
131      *      a[i] == !a[i-1]
132      *              == !((i-1)%2==0)
133      *              == !false
134      *              == true
135      *              == (i%2==0).
136      *
137      * vero nel punto (B) che e' esattamente P(i). */
138 }
139
```