RJEŠENJA:

1. Stupanj liste je 4
   $h \le 1 + \log_2 8 = 1 + 3 = 4$

   Očekivani brojevi elemenata po stupnjevima su:
   $E(n_1) = 8 \cdot (1/2)^0 \cdot 1/2 = 4$
   $E(n_2) = 8 \cdot (1/2)^1 \cdot 1/2 = 2$
   $E(n_3) = 8 \cdot (1/2)^2 \cdot 1/2 = 1$
   $E(n_4) = 8 \cdot (1/2)^3 \cdot 1/2 = ½$ ->1

   Najpraktičnije je odmah skalirati zadane brojeve na interval širine 8 (npr. U[0,8]). Zatim možemo provesti pretvorbu u cijele brojeve ili jednostavno odrediti intervale pretinaca unutar naše razdiobe pritom poštujući širine pretinaca.

   a) Pretinci bi u ovom slučaju bili: [0,4);[4,6);[6,7);[7,8] (granični elementi proizvoljno raspoređeni)
   b) U slučaju pretvorbe u ciljani interval cijelih brojeva [1,8] treba paziti da je svaki broj jednako vjerojatan (tj. da dobijemo diskretnu uniformnu razdiobu U[1,8]). Ako su brojevi izvorno iz kontinuiranog intervala X~U[0,8], pretvorba u diskretne se može obaviti po formuli

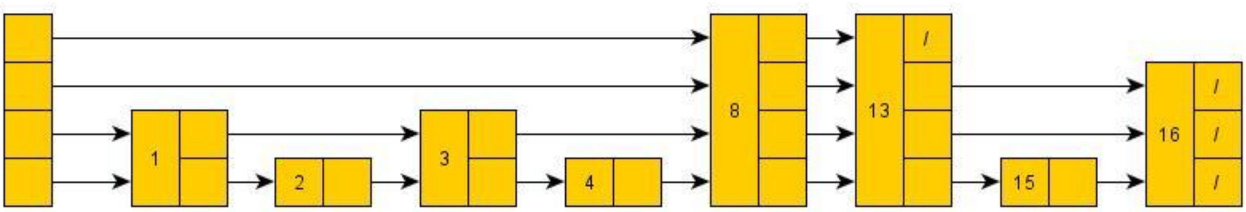
   $$Y = \begin{cases} 8, & za\ X = 8 \\ \lfloor X + 1 \rfloor, & inače \end{cases}.$$

   Pretinci su: [1,4]; [5,6]; [7,7]; [8,8]

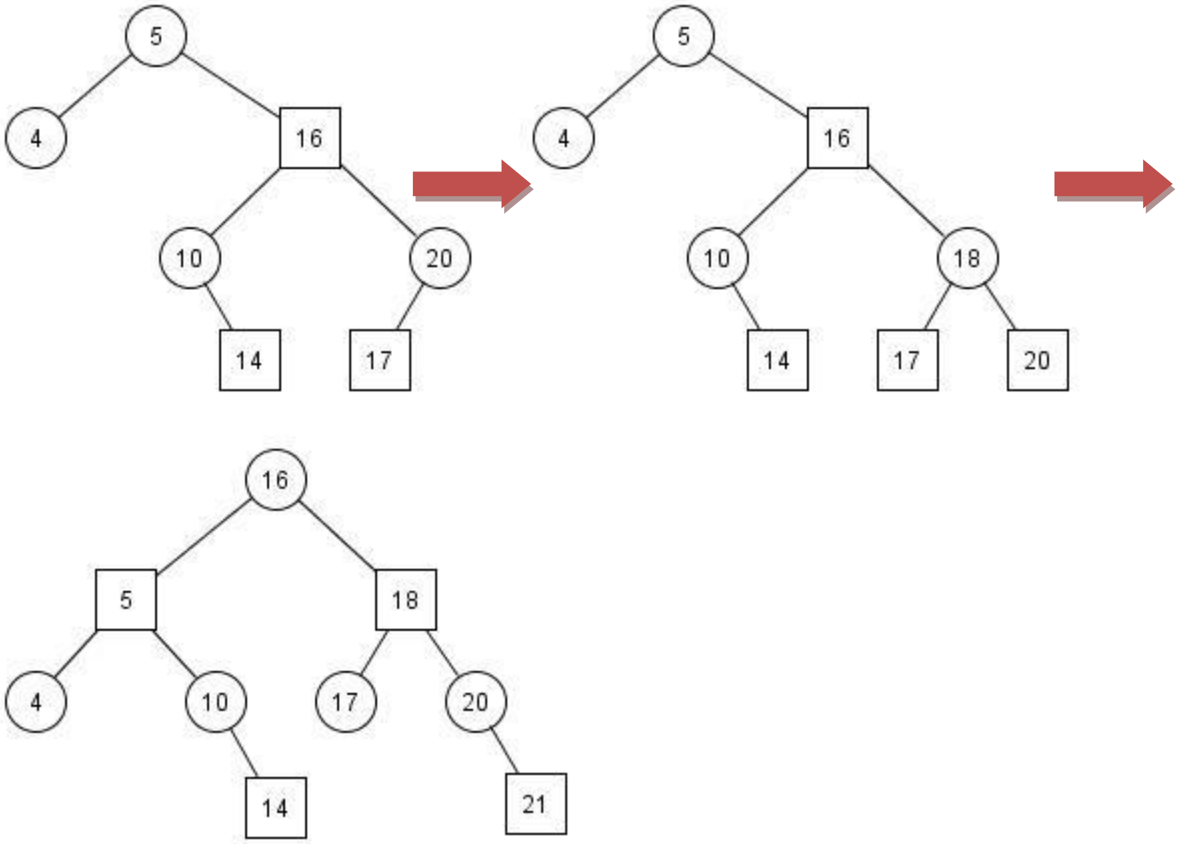
   c) Naravno, možemo i interval [0,1] podijeliti odmah na pretince u omjerima koje nalažu brojevi elemenata po stupnjevima.

   U svakom slučaju, slučajni brojevi određuju sljedeće pretince:

X	0.81	0.99	0.69	0.91	0.68	0.42	0.09	0.18	0.13
brojevi	16	8	3	13	1	15	2	4	
pretinac	3.	4.	2.	4.	2.	1.	1.	1.	1.



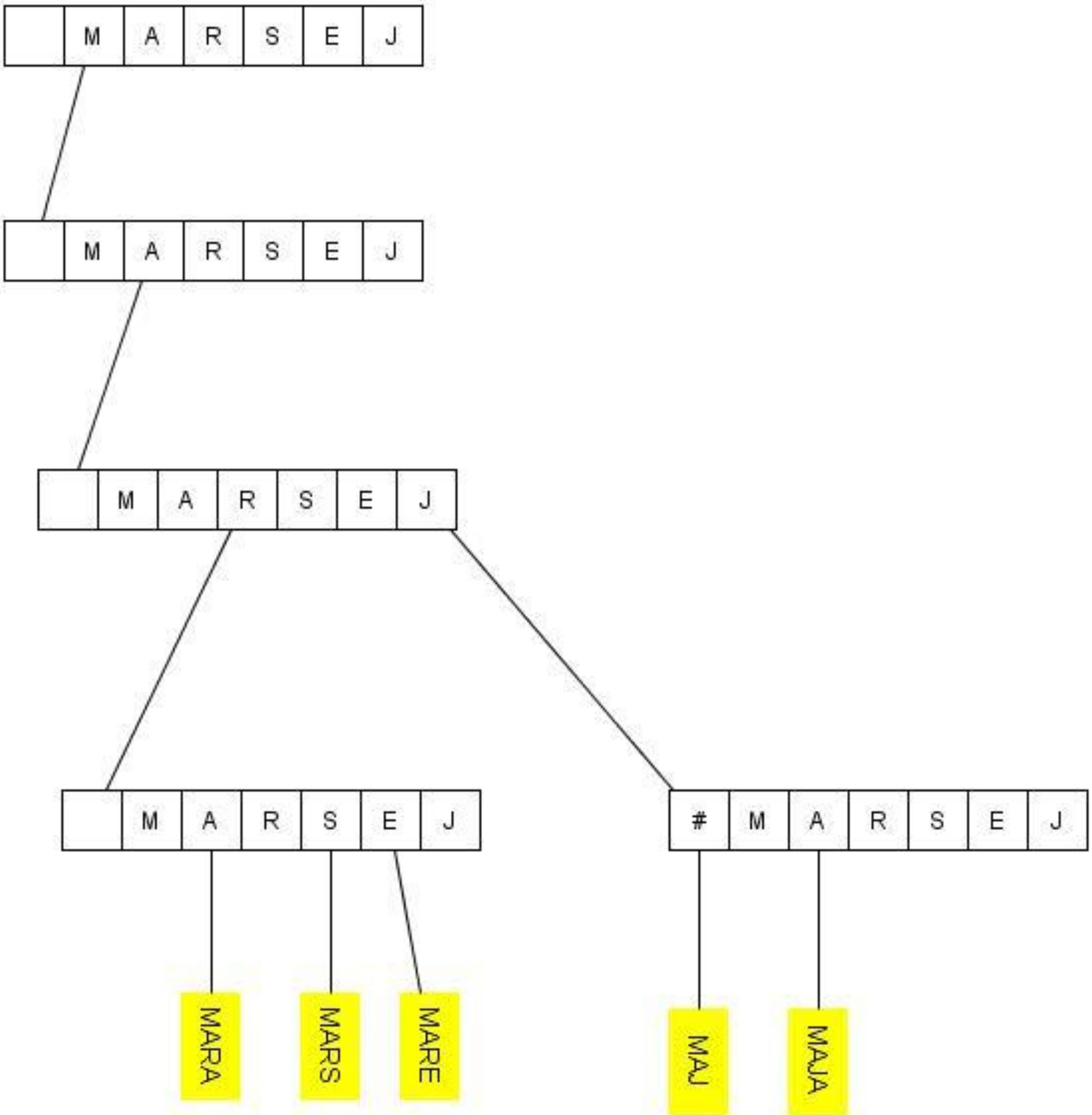
2.



3.

```
SearchBTree (key, node)

        if (node != NULL)

        {

        //keyNum je član čvora

                for (i=1; i<=node->keyNum && node->keys[i]<key; ++i);

                if (i>node->keyNum || node->keys[i]>key)

                        SearchBTree (key, node->pointers[i]);

                else

                        return node;}

        else

        no searched key;
```

4.



5.

Jedinka	10110	01111	10101	10111	00001
dobrota	66	39	27	15	3

p_k=0.8
p_m=0.1.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
ω	0.2	0.84	0.05	0.41	0.22	0.62	0.21	0.19	0.44	0.06	0.54	0.76	0.48	0.01

Slučajni brojevi se koriste redom:
ω(1), ω(2) za odabir jedinki
ω(3) za odluku o križanju: 0.05<0.8 -> križanja će biti
ω(4) odabir točke prekida
ω(5)- ω(14): odluka o mutaciji na obje dobivene jedinke, ispitivanje na svakom bitu.

razmjerni odabir:
vjerojatnosti odabira pojedinih jedinki:
p_1=66/150=0.44
p_2=39/150=0.26
p_3=27/150=0.18
p_4=15/150=0.1
p_5=3/150=0.02

Odabrane su 1. i 3. jedinka.

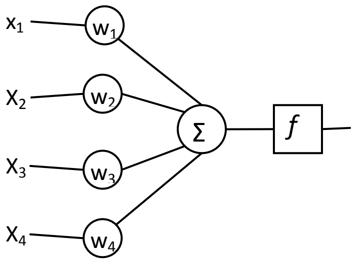
Točka prekida: $\lfloor 4 * 0.41 + 1 \rfloor = 2$ (4 su moguće točke prekida, tj. sve moguće pozicije između gena).
Određivanje točke prekida se može obaviti i pretincima.

Dobivene jedinke:
10101
10110

Mutacija:
Za izvođenje mutacije na prvoj jedinci se koriste ω(5)- ω(9) te ne dolazi ni do jedne mutacije
Za izvođenje mutacije na drugoj jedinci se koriste ω(10)- ω(14) te dolazi do mutacije na prvom i zadnjem genu.

Rezultat:
10101
00111

6. a) $f(z) = z$



b) $p = n = 4 \quad \Rightarrow$ ako postoji, rješenje je jedinstveno

c)

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \qquad \mathbf{x}_d = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \mathbf{y}_d = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad ; \qquad \mathbf{A} = \mathbf{x}_d^T,\ \mathbf{b}=\mathbf{y},\ \text{sustav } \mathbf{Aw} = \mathbf{b}$$

$\mathbf{X}_d^T = \mathbf{X}_d$

$$\Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \Leftrightarrow \mathbf{I}_4 \cdot \mathbf{w} = \mathbf{b}$$

$$\Rightarrow \quad \mathbf{w} = \mathbf{b} \quad \Rightarrow \quad \mathbf{w} = [1\ 0\ 0\ 1]^T$$

e) LMS

k=0: $\mathbf{w}^{(0)} = [0\ 0\ 0\ 0]^T,\ \alpha = 1$

k=1: $e_1^{(0)} = \mathbf{x}^T_{d,1}\mathbf{w}^{(0)} - y_{d,1} = [1\ 0\ 0\ 0] \cdot [0\ 0\ 0\ 0]^T - 1 = -1$

$\mathbf{w}^{(1)} = \mathbf{w}^{(0)} - \alpha \cdot \mathbf{x}_{d,1} \cdot e_1^{(0)} = [0\ 0\ 0\ 0]^T - 1 \cdot [1\ 0\ 0\ 0]^T \cdot (-1) = [1\ 0\ 0\ 0]^T$

k=2: $e_2^{(1)} = \mathbf{x}^T_{d,2}\mathbf{w}^{(1)} - y_{d,2} = [0\ 0\ 0\ 1] \cdot [1\ 0\ 0\ 0]^T - 1 = -1$

$\mathbf{w}^{(2)} = \mathbf{w}^{(1)} - \alpha \cdot \mathbf{x}_{d,2} \cdot e_2^{(1)} = [1\ 0\ 0\ 0]^T - 1 \cdot [0\ 0\ 0\ 1]^T \cdot (-1) = [1\ 0\ 0\ 1]^T$

k=3: $e_3^{(2)} = \mathbf{x}^T_{d,3}\mathbf{w}^{(2)} - y_{d,3} = [0\ 0\ 1\ 0] \cdot [1\ 0\ 0\ 1]^T - 0 = 0$

$\mathbf{w}^{(3)} = \mathbf{w}^{(2)}$

U ovom slučaju, LMS algoritam pronalazi rješenje već nakon prva dva koraka prve iteracije (epohe).