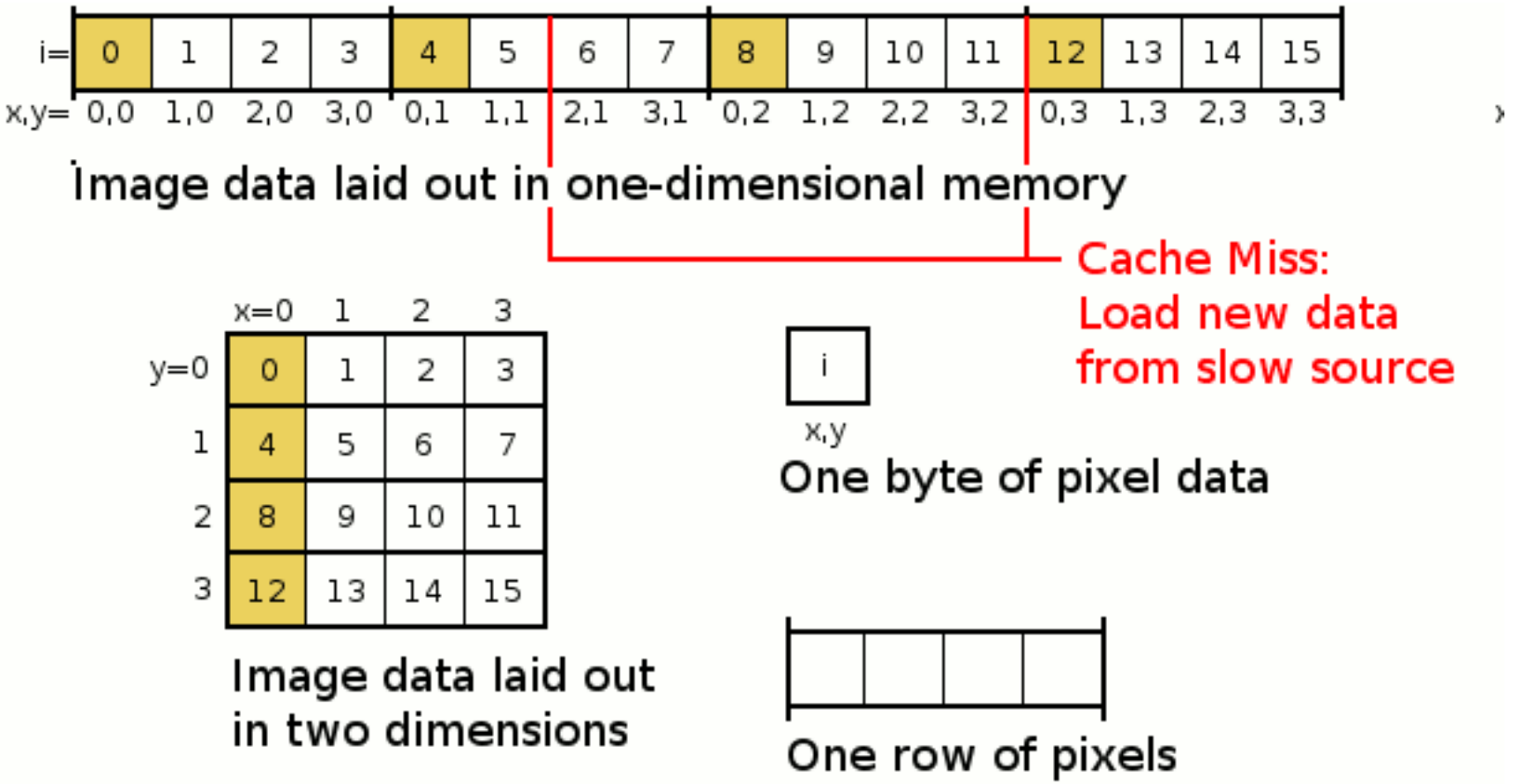
Optimiranje memorijskog rasporeda “offline” binarnog stabla

Student: Sandi Fatić
Voditelj: Siniša Šegvić
Siječanj 2015.

Optimiranje rasporeda obilaženja matrice



Stablo I memorija

Dinamički alocirano stablo (new ili malloc)

Statički alocirano stablo (statički blok memorije)

Locality

Temporal locality

Spatial locality

Dinamički alocirano stablo

1) Raspored memorije ovisi o ubacivanjima

2) Stablo nije u kompaktnom bloku memorije

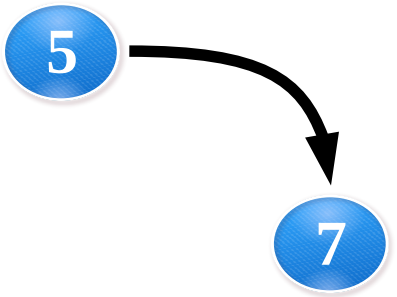
3) new (malloc) overhead

Dinamički alocirano stablo

5					

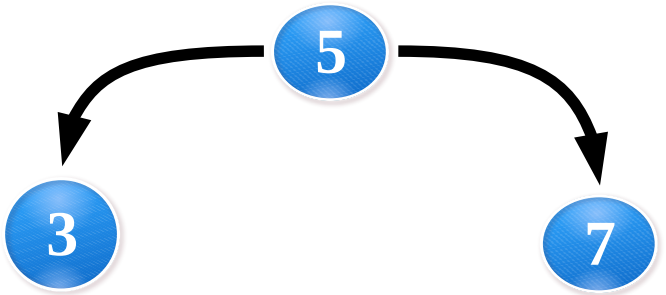
Dinamički alocirano stablo

5	7				



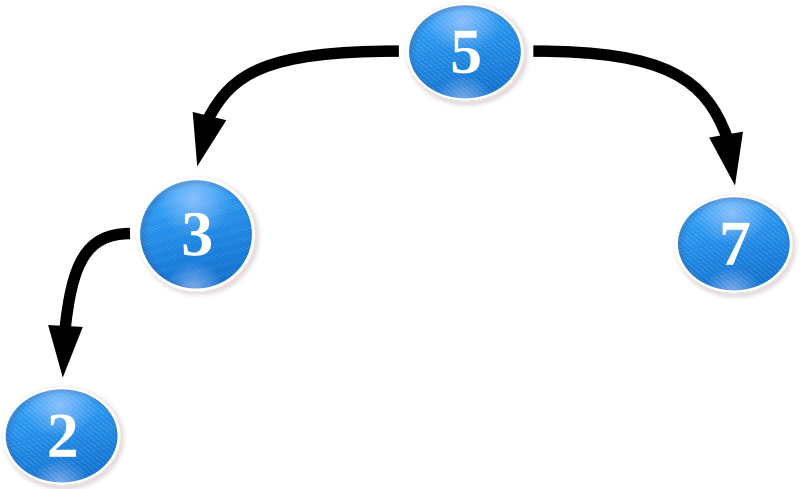
Dinamički alocirano stablo

5	7	3				



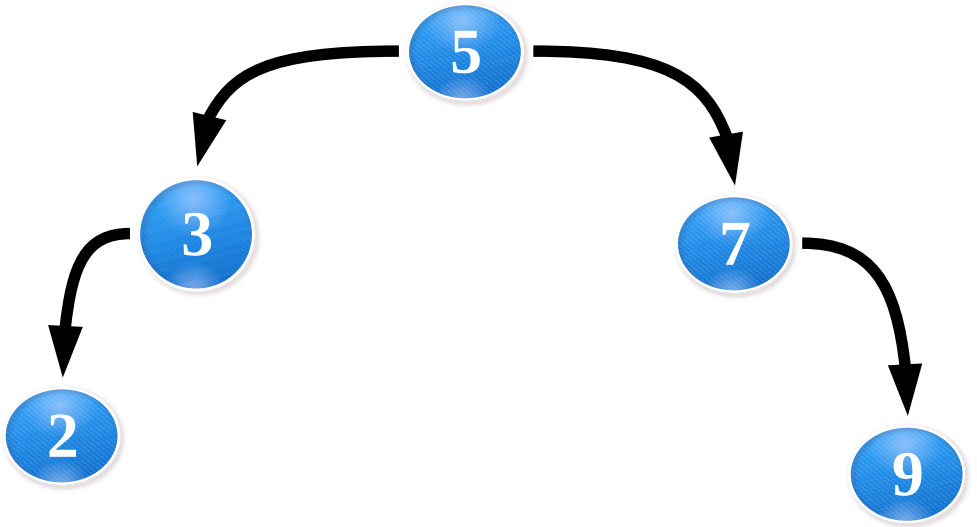
Dinamički alocirano stablo

5	7	3	2			



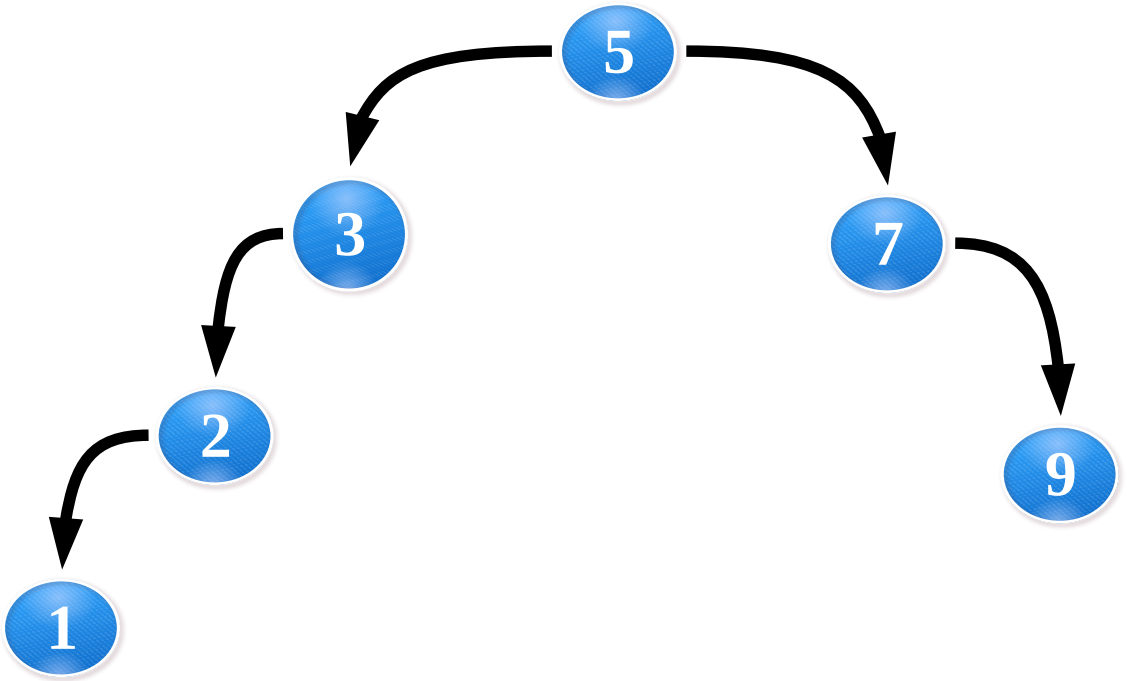
Dinamički alocirano stablo

5	7	3	2	9			



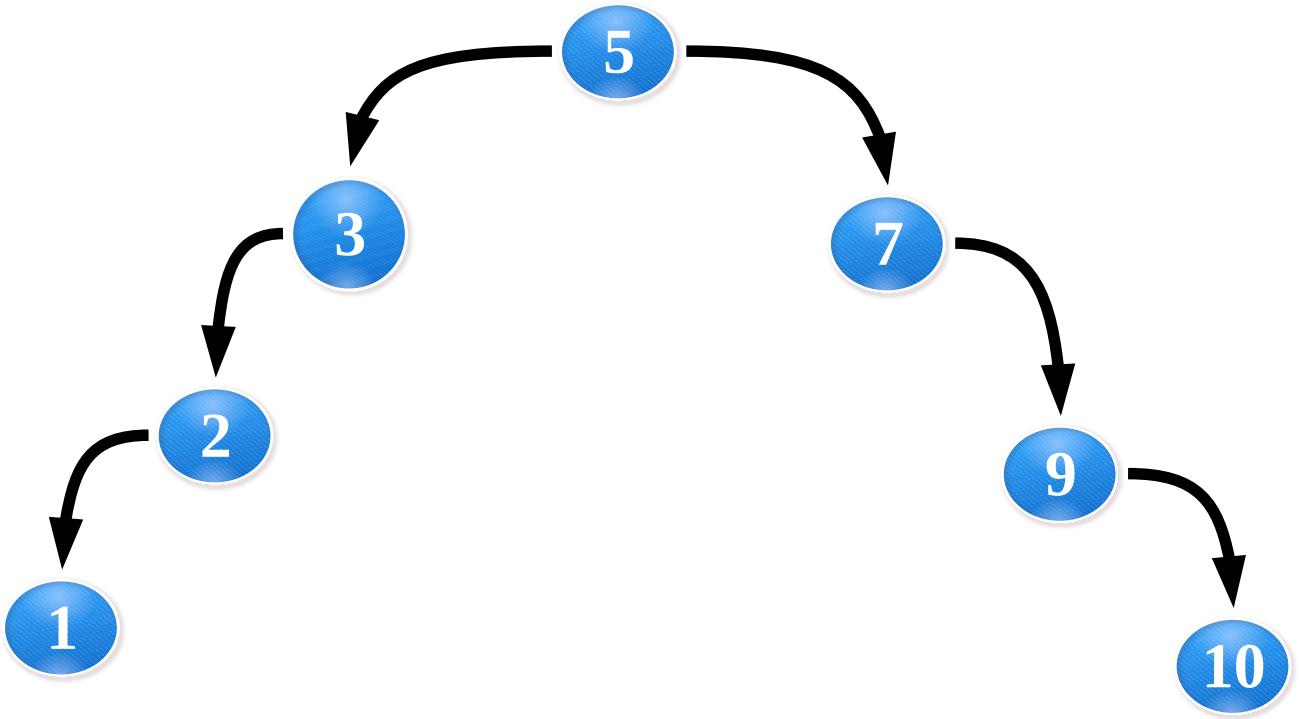
Dinamički alocirano stablo

5	7	3	2	9	1		



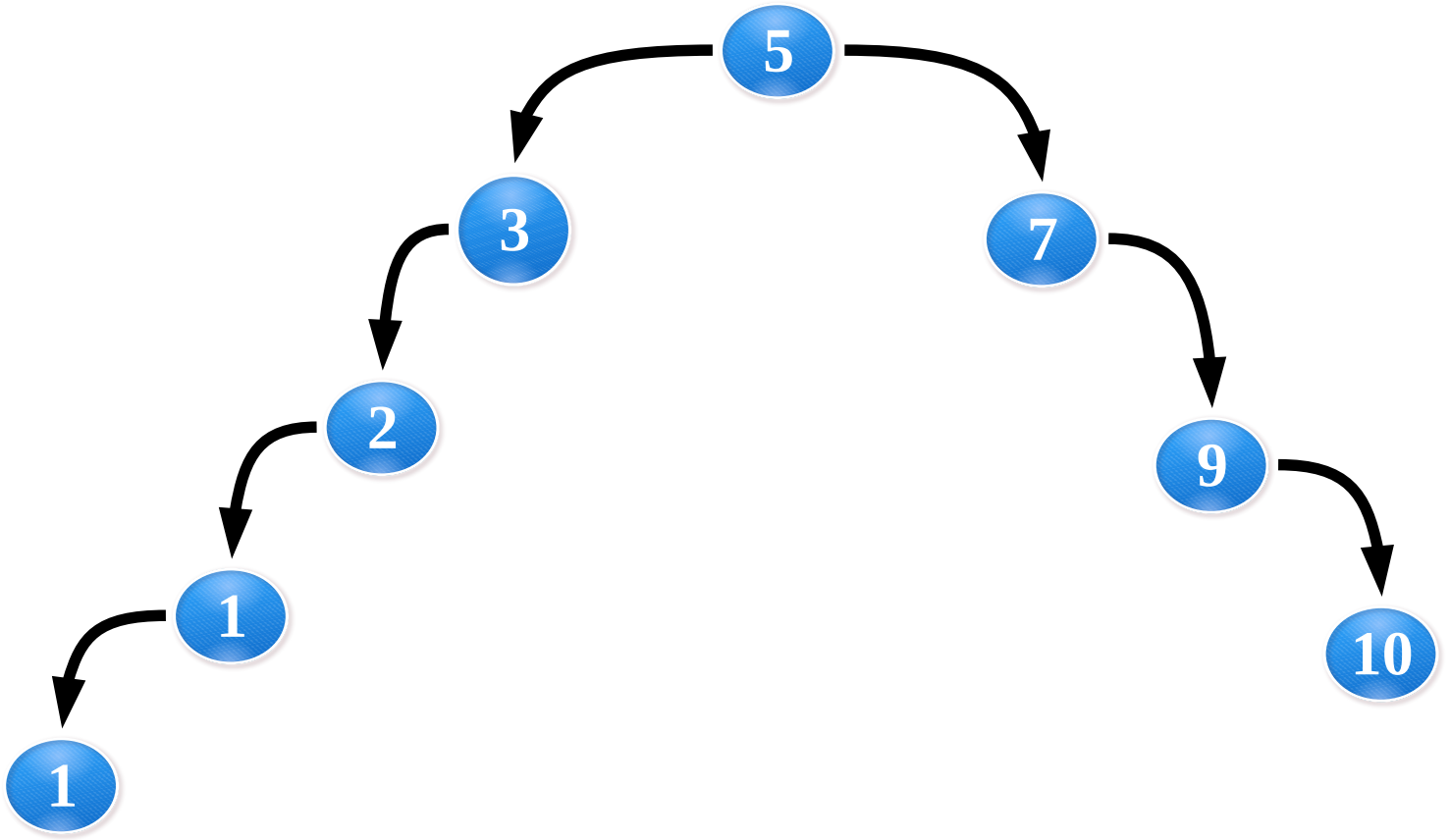
Dinamički alocirano stablo

5	7	3	2	9	1	10		



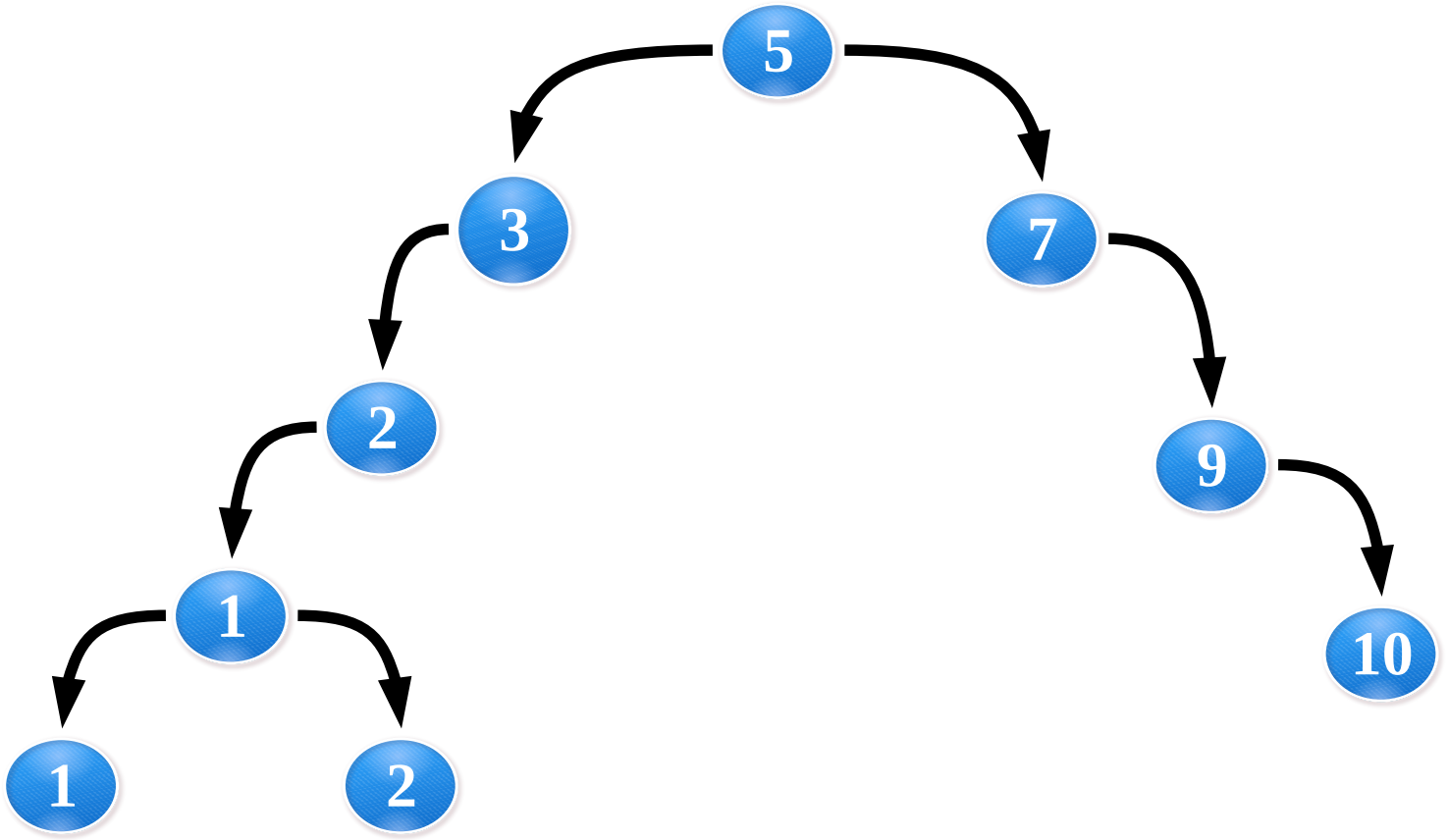
Dinamički alocirano stablo

5	7	3	2	9	1	10	1	



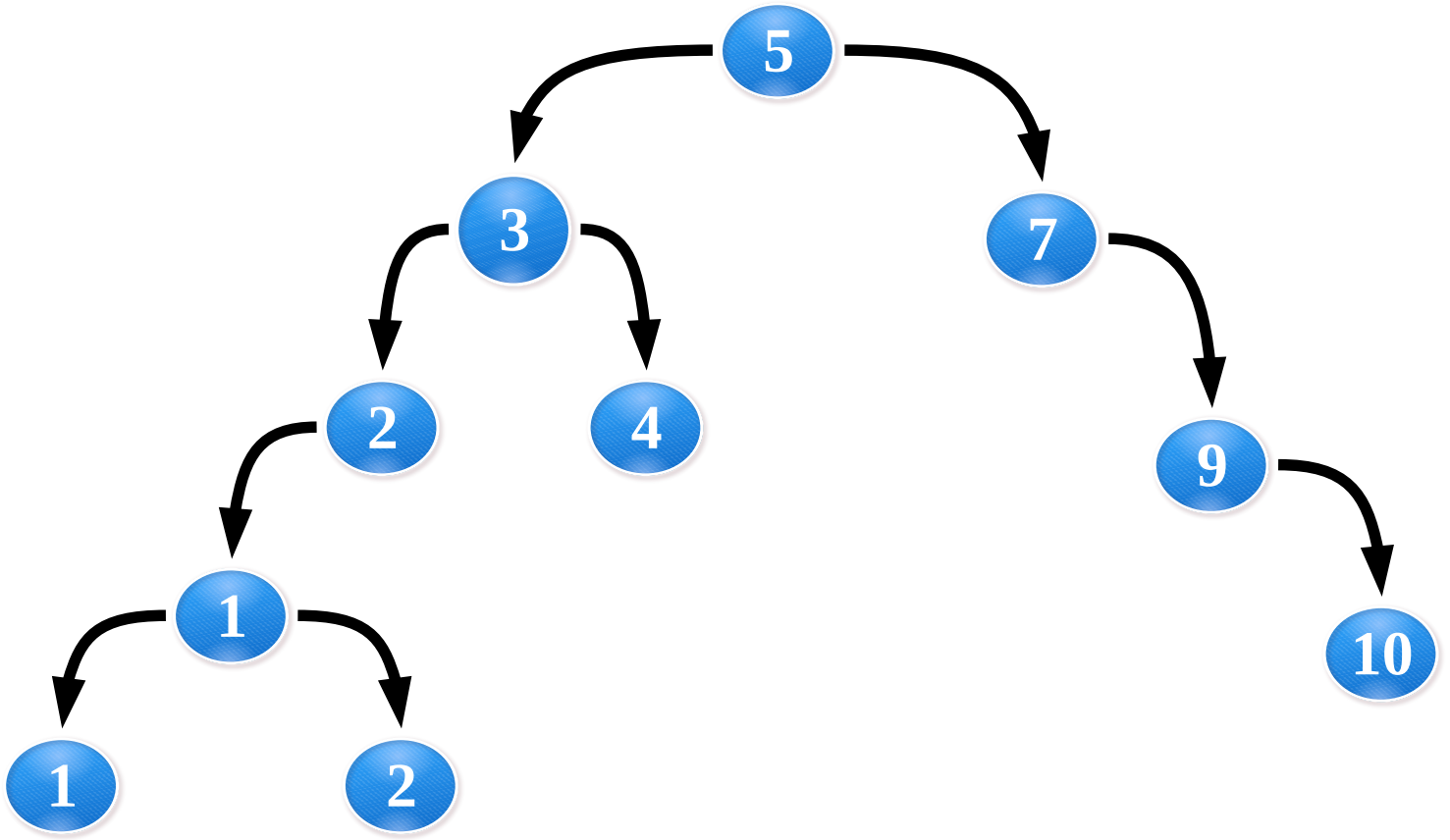
Dinamički alocirano stablo

5	7	3	2	9	1	10	1	2	



Dinamički alocirano stablo

5	7	3	2	9	1	10	1	2	4



Dinamički alocirano stablo

```
72 template <class T>
73 void BinaryTree<T>::insertElement(BinaryNode<T>* current, const T& elem) {
74     if ((*current).data < elem)
75     {
76         if (current->right == NULL)
77         {
78             current->right = new BinaryNode<T>(elem);
79             return;
80         }
81
82         insertElement(current->right, elem);
83     }
84     else
85     {
86         if (current->left == NULL)
87         {
88             current->left = new BinaryNode<T>(elem);
89             return;
90         }
91
92         insertElement(current->left, elem);
93     }
94 }
95
```

Statički alocirano stablo

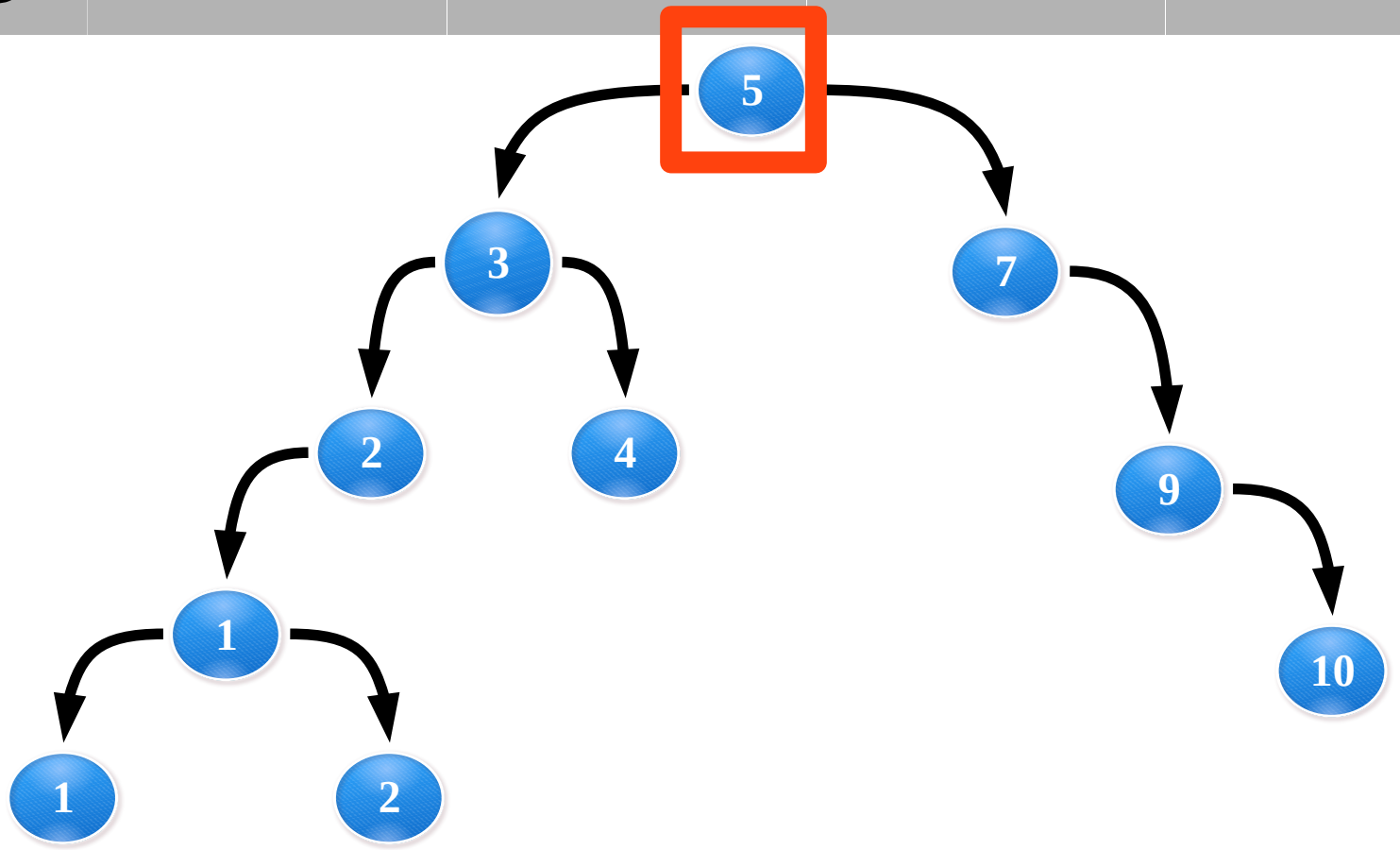
1) Blok memorije (spatial locality)

2) Raspored elemenata (spatial locality)

3) Nema new (malloc) overhead-a

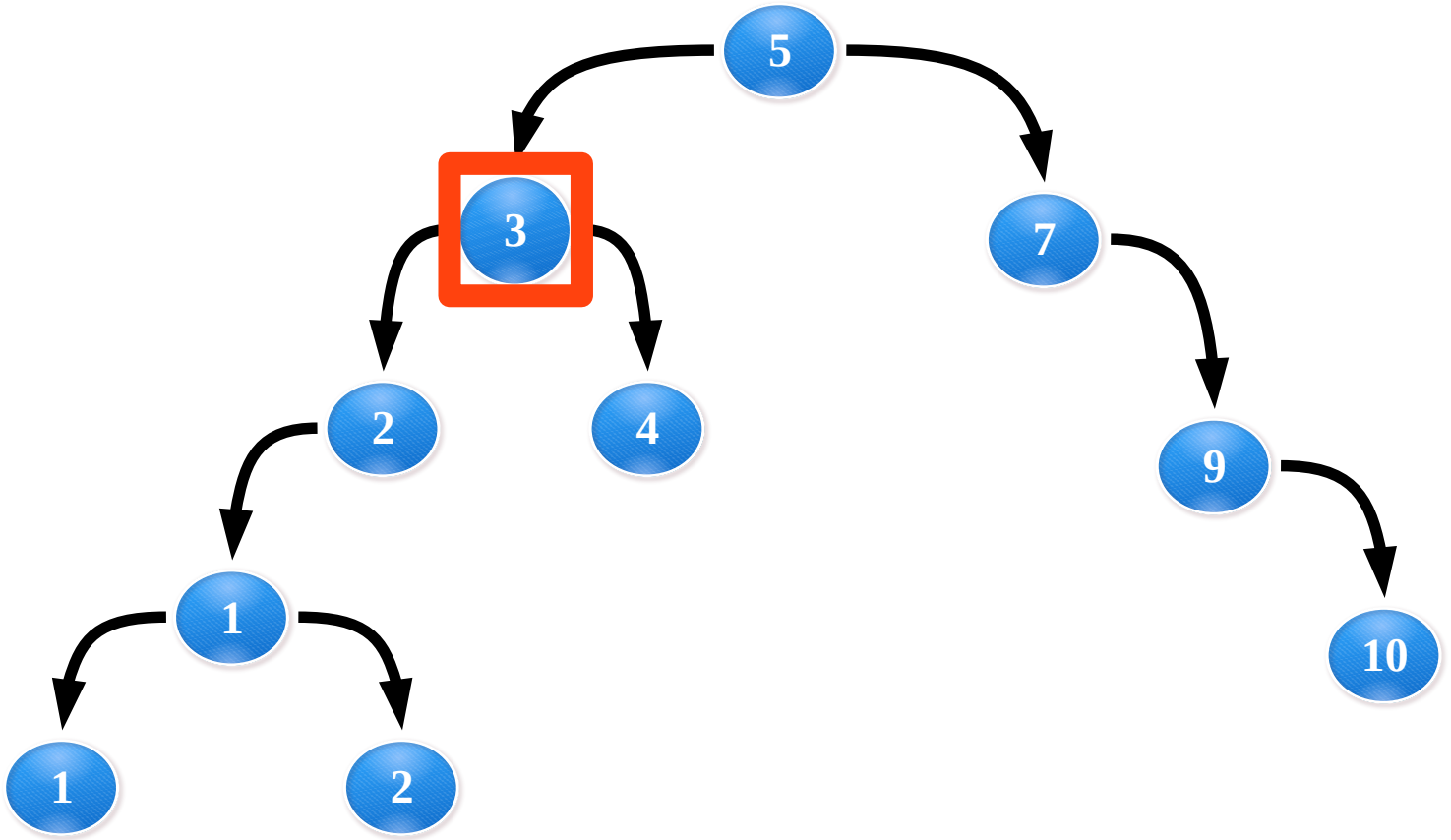
Statički alocirano stablo (DFS lijevo)

5					



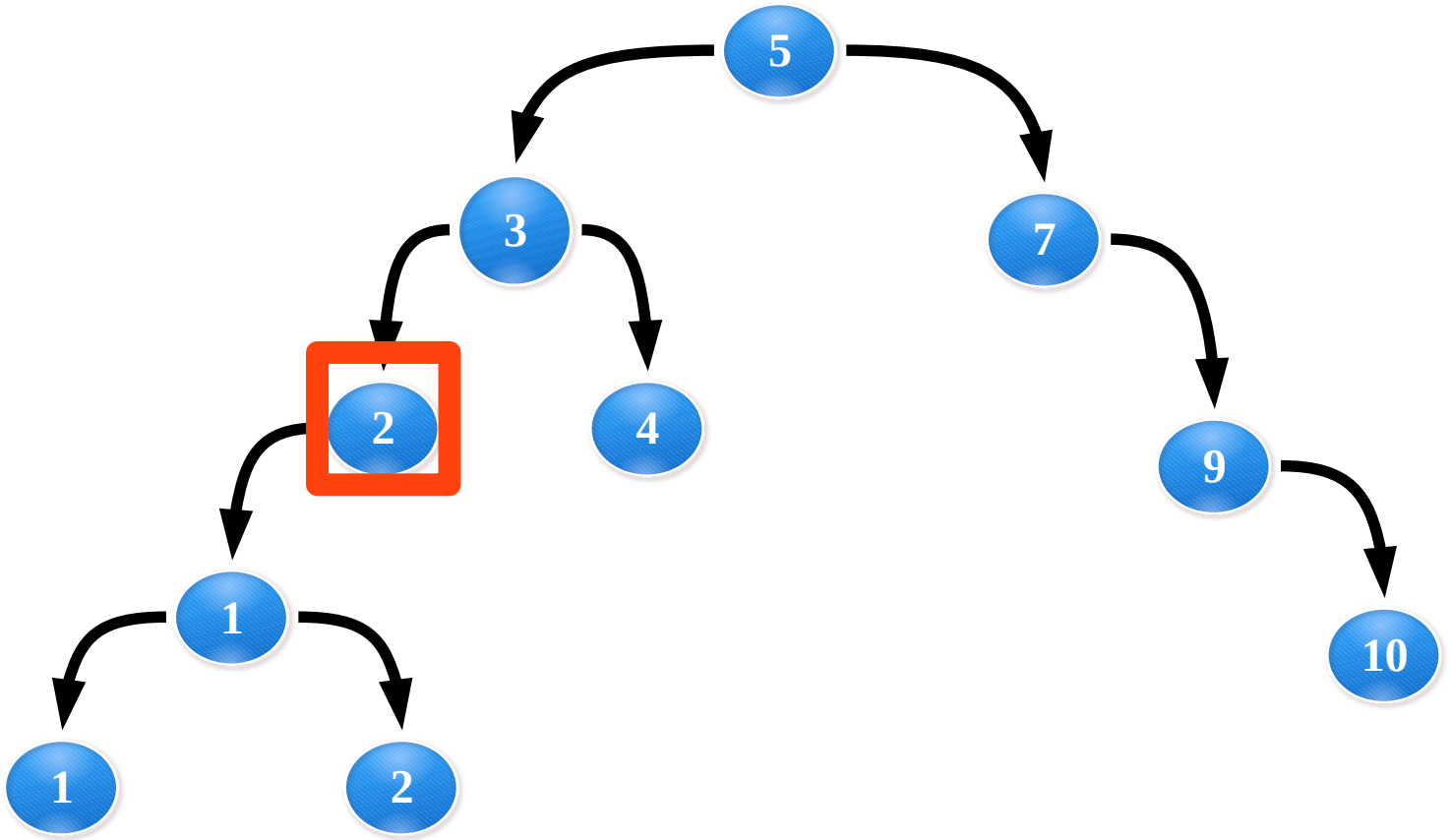
Statički alocirano stablo (DFS lijevo)

5	3				



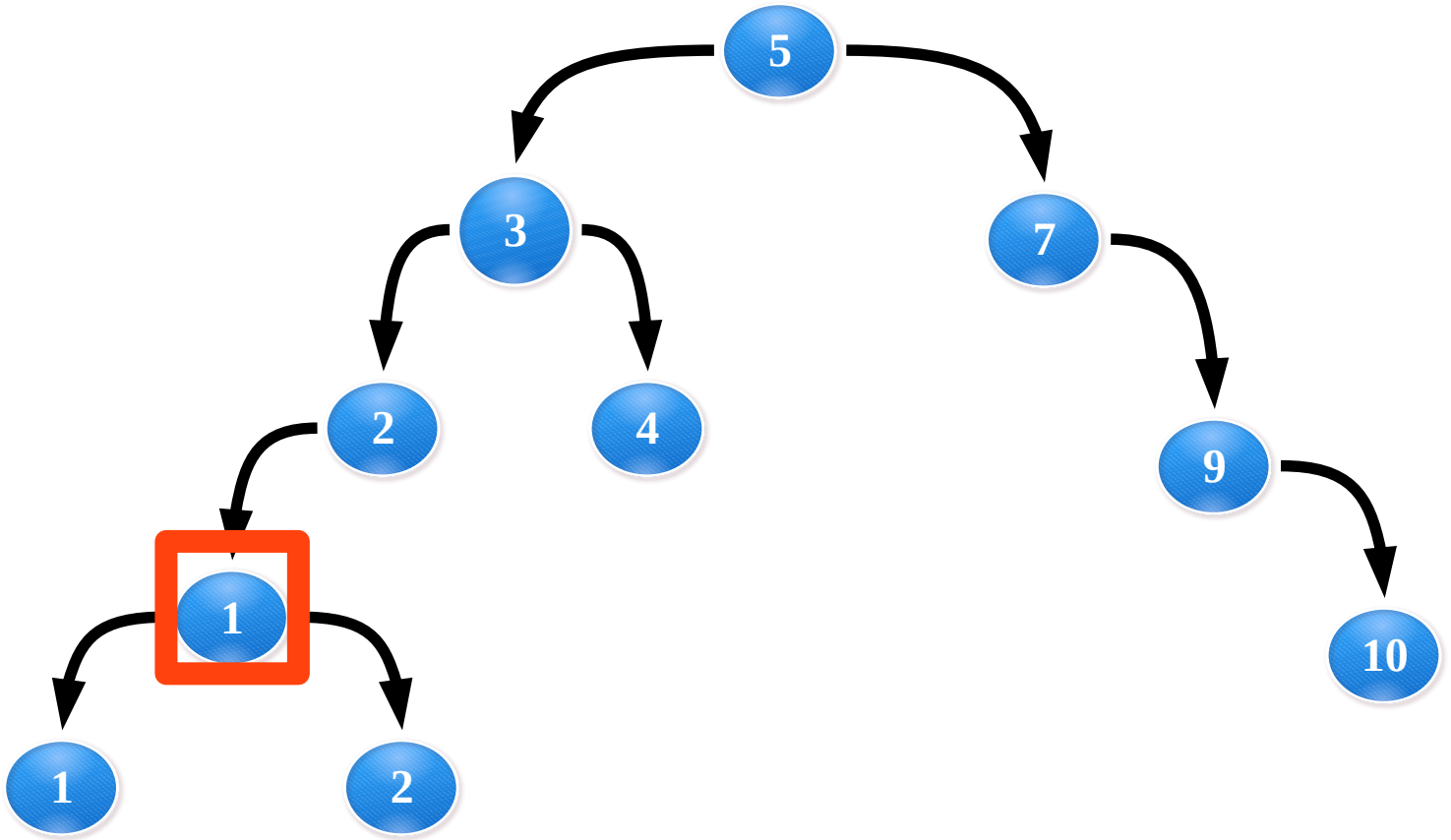
Statički alocirano stablo (DFS lijevo)

5	3	2				



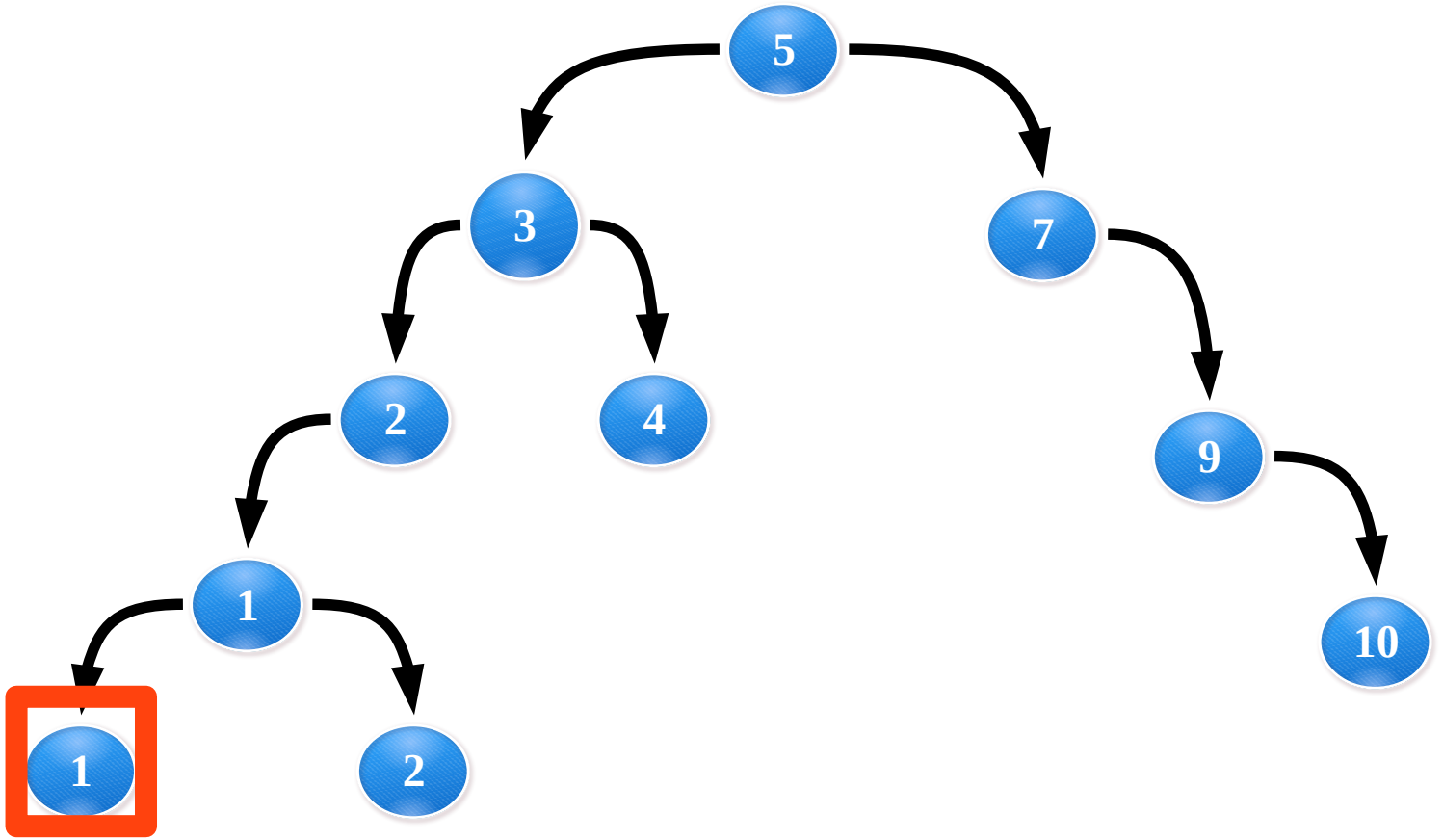
Statički alocirano stablo (DFS lijevo)

5	3	2	1			



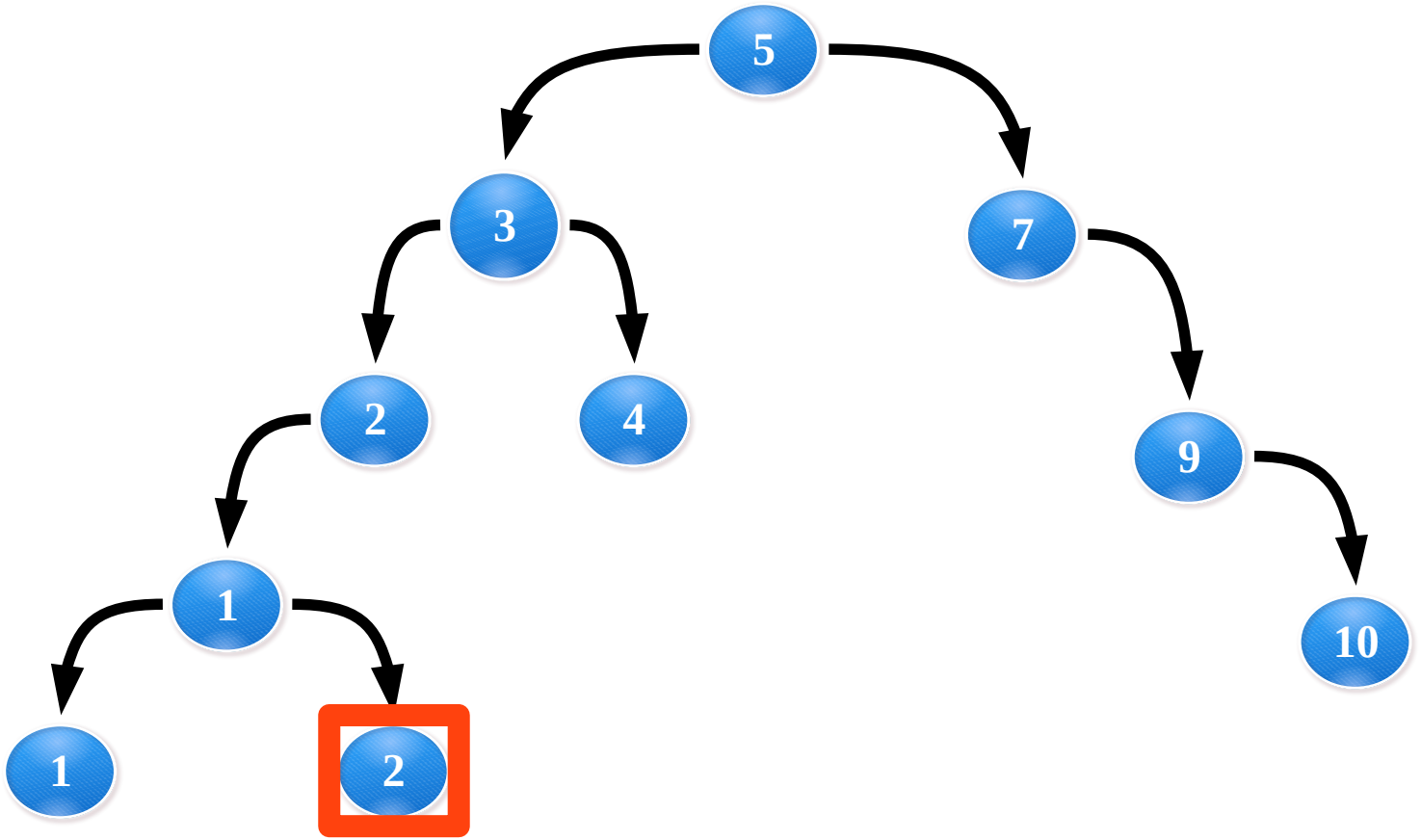
Statički alocirano stablo (DFS lijevo)

5	3	2	1	1			



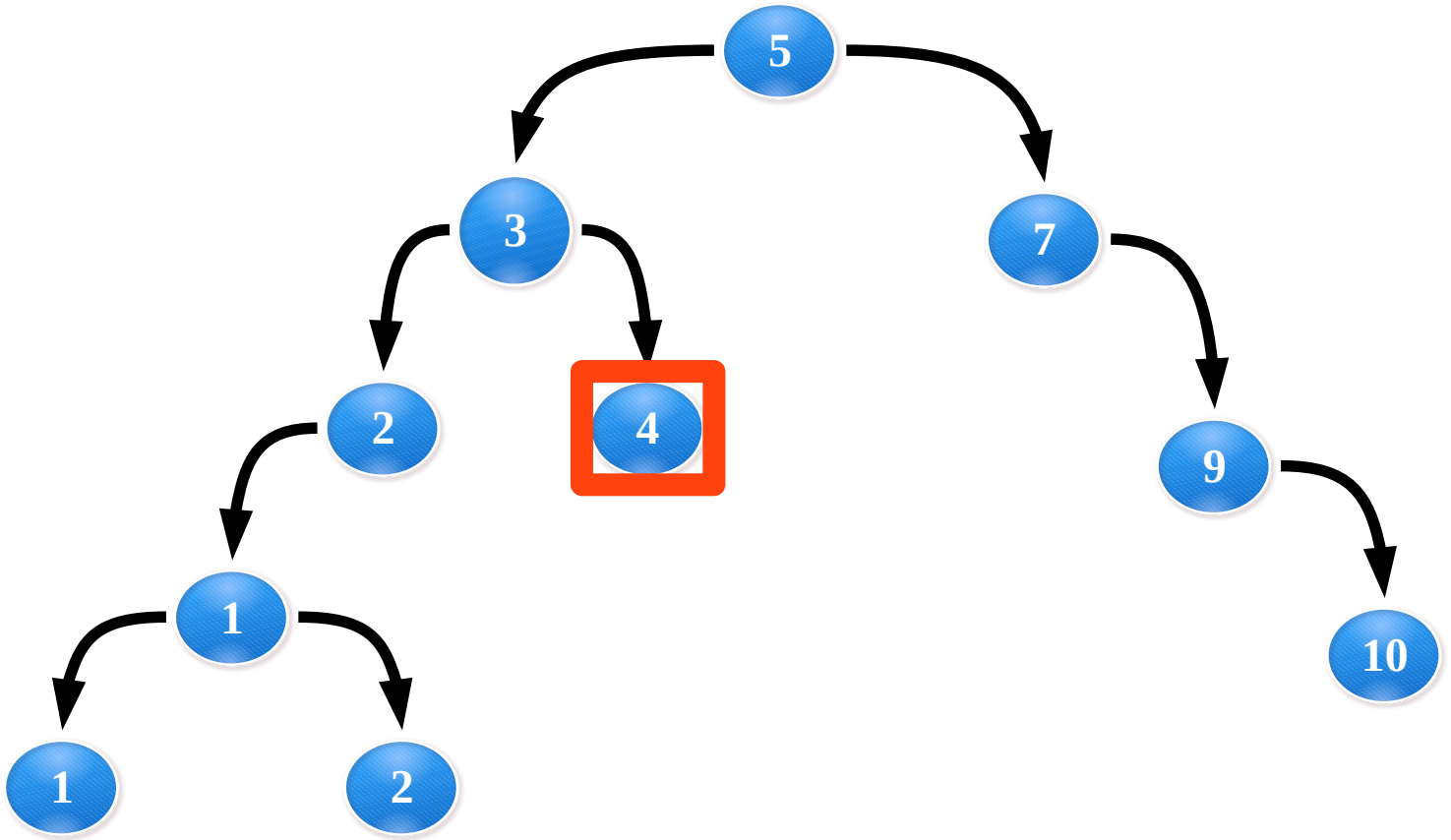
Statički alocirano stablo (DFS lijevo)

5	3	2	1	1	2		



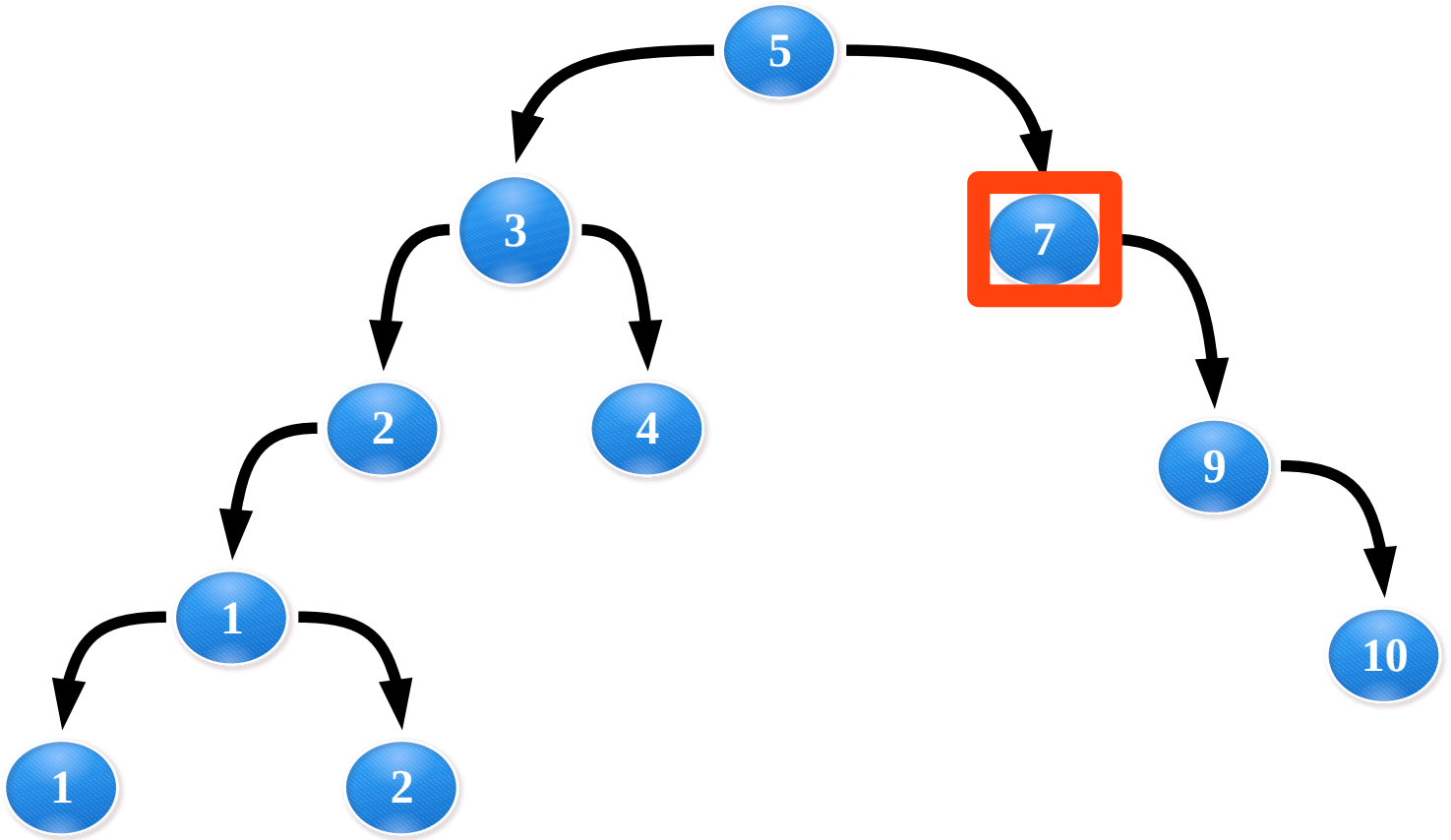
Statički alocirano stablo (DFS lijevo)

5	3	2	1	1	2	4		



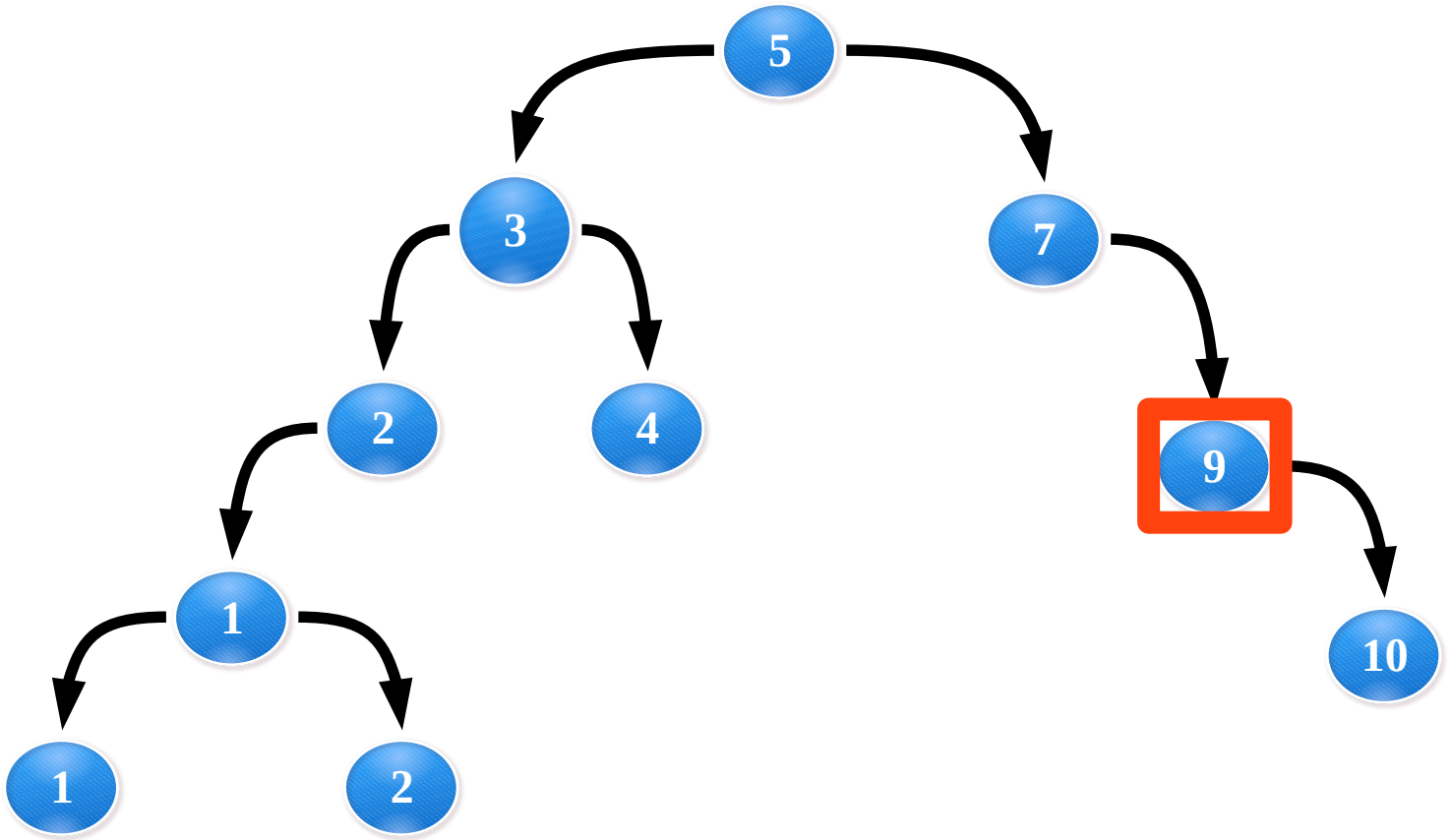
Statički alocirano stablo (DFS lijevo)

5	3	2	1	1	2	4	7	



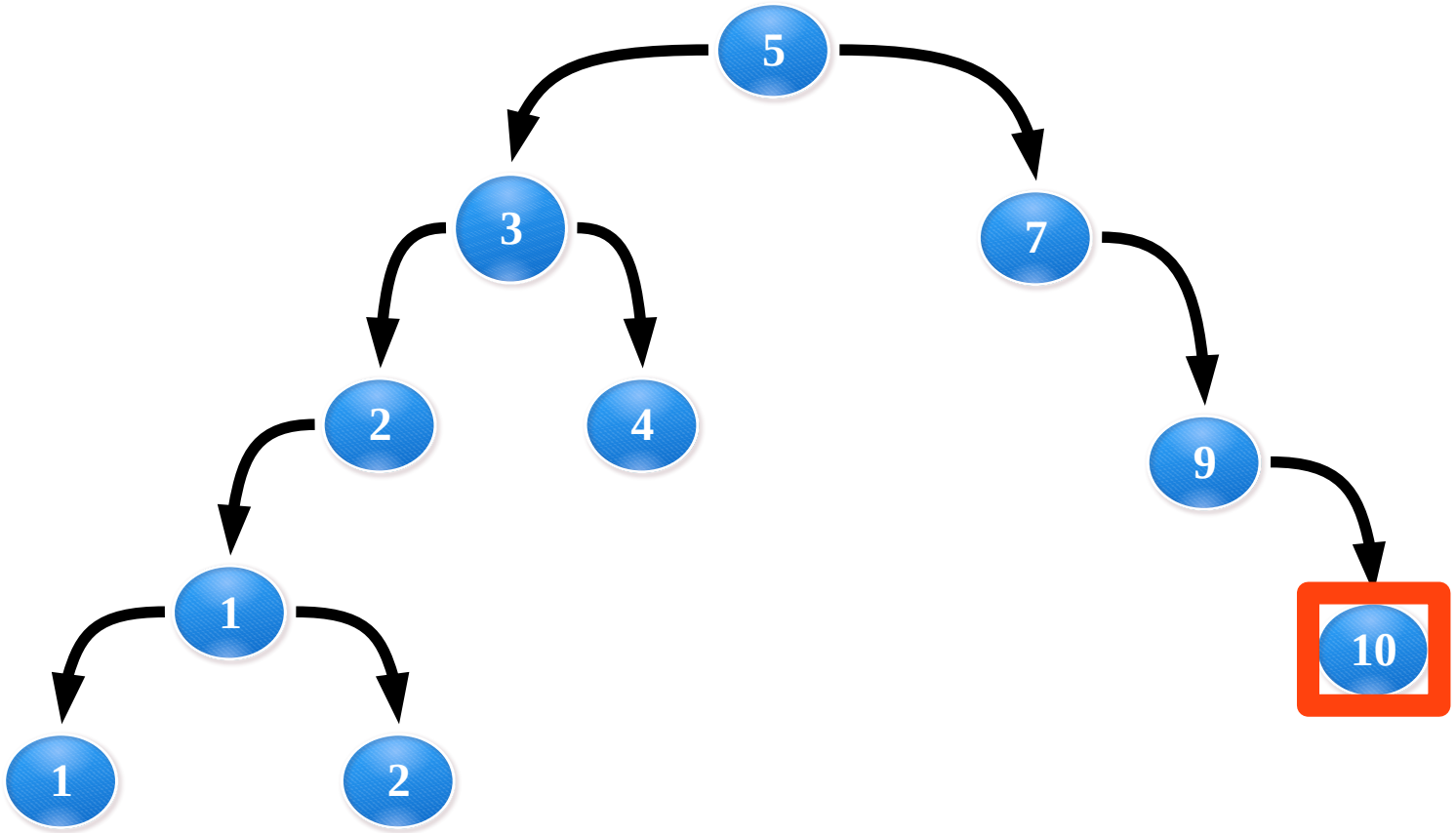
Statički alocirano stablo (DFS lijevo)

5	3	2	1	1	2	4	7	9	



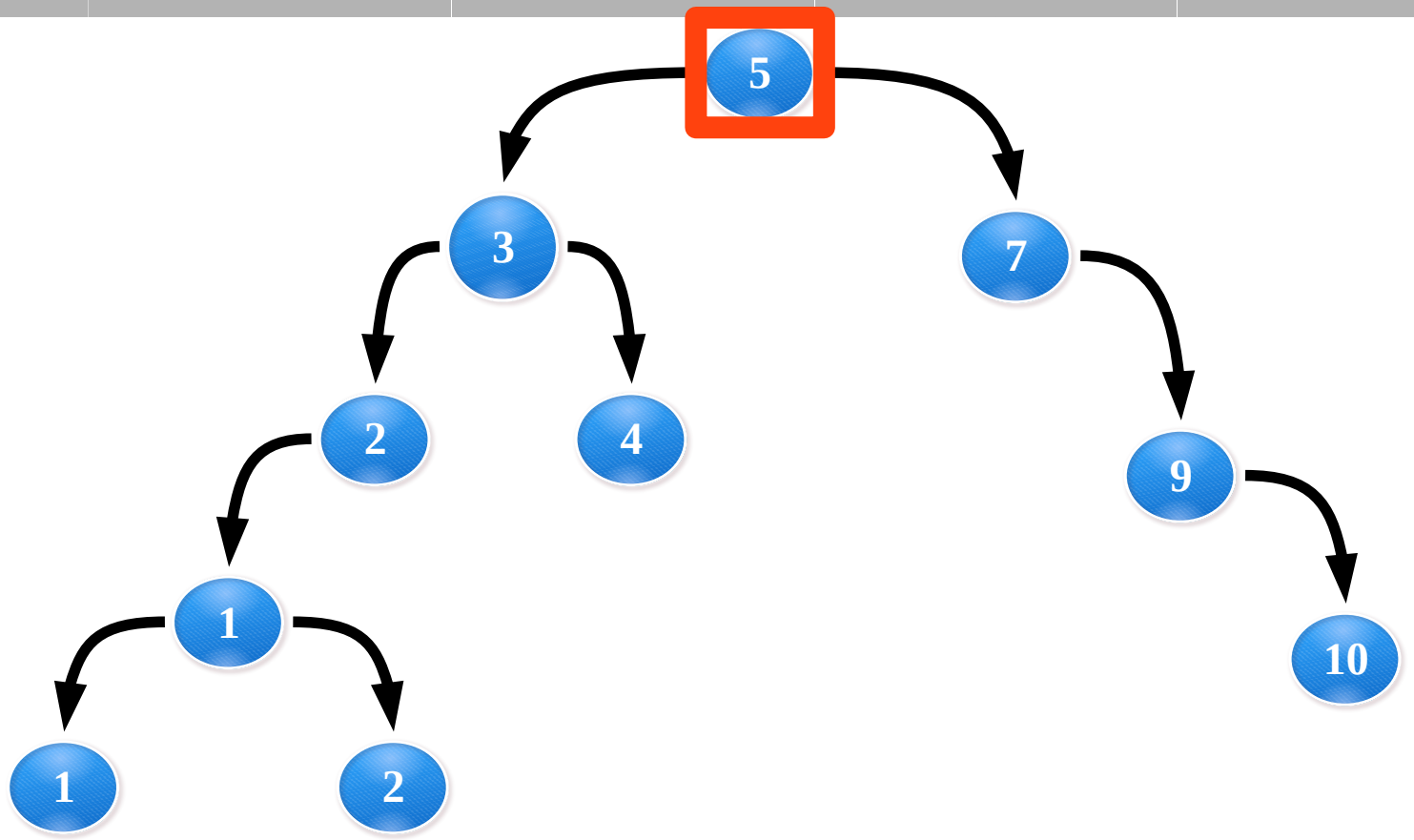
Statički alocirano stablo (DFS lijevo)

5	3	2	1	1	2	4	7	9	10



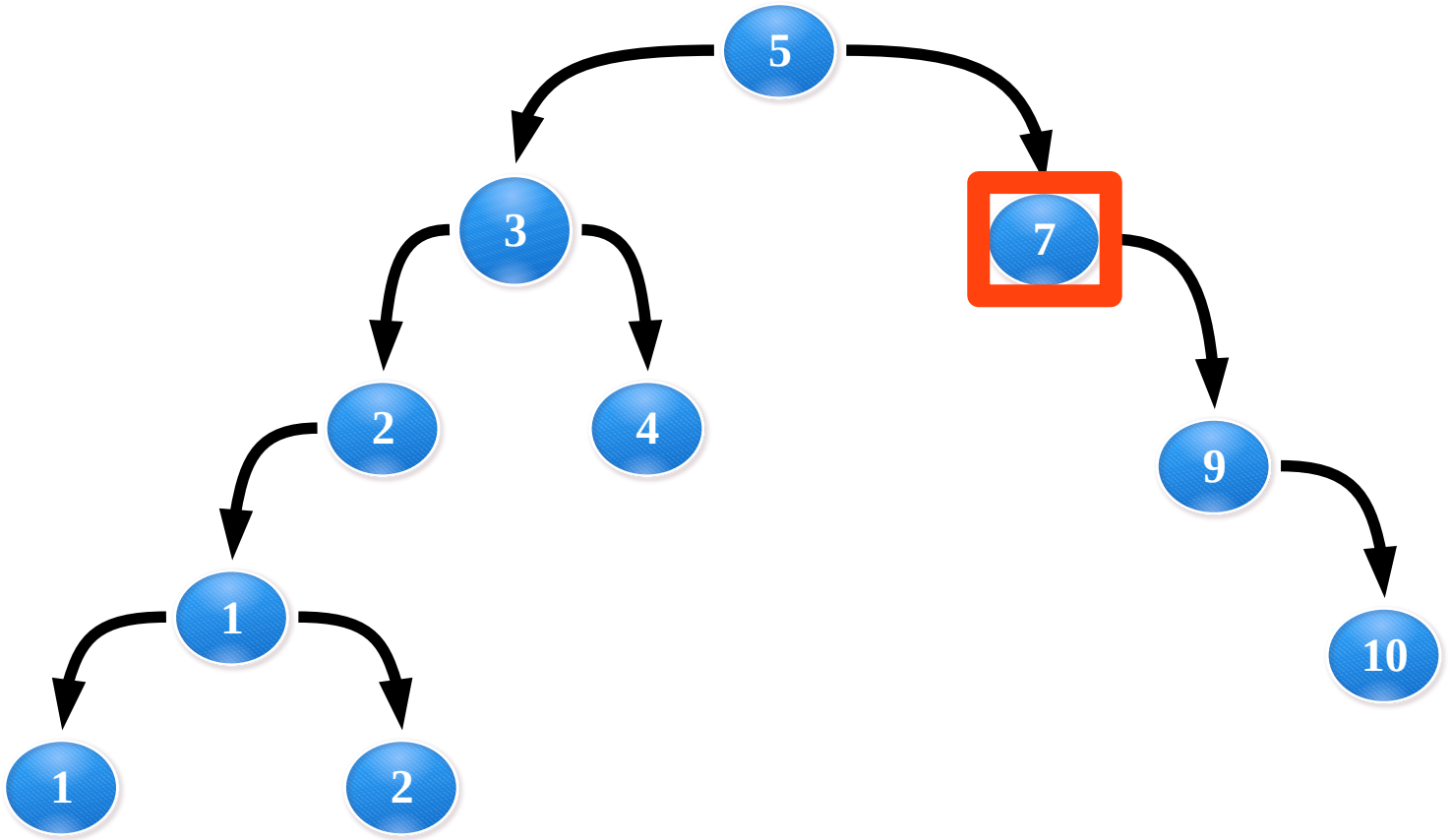
Statički alocirano stablo (DFS desno)

5					



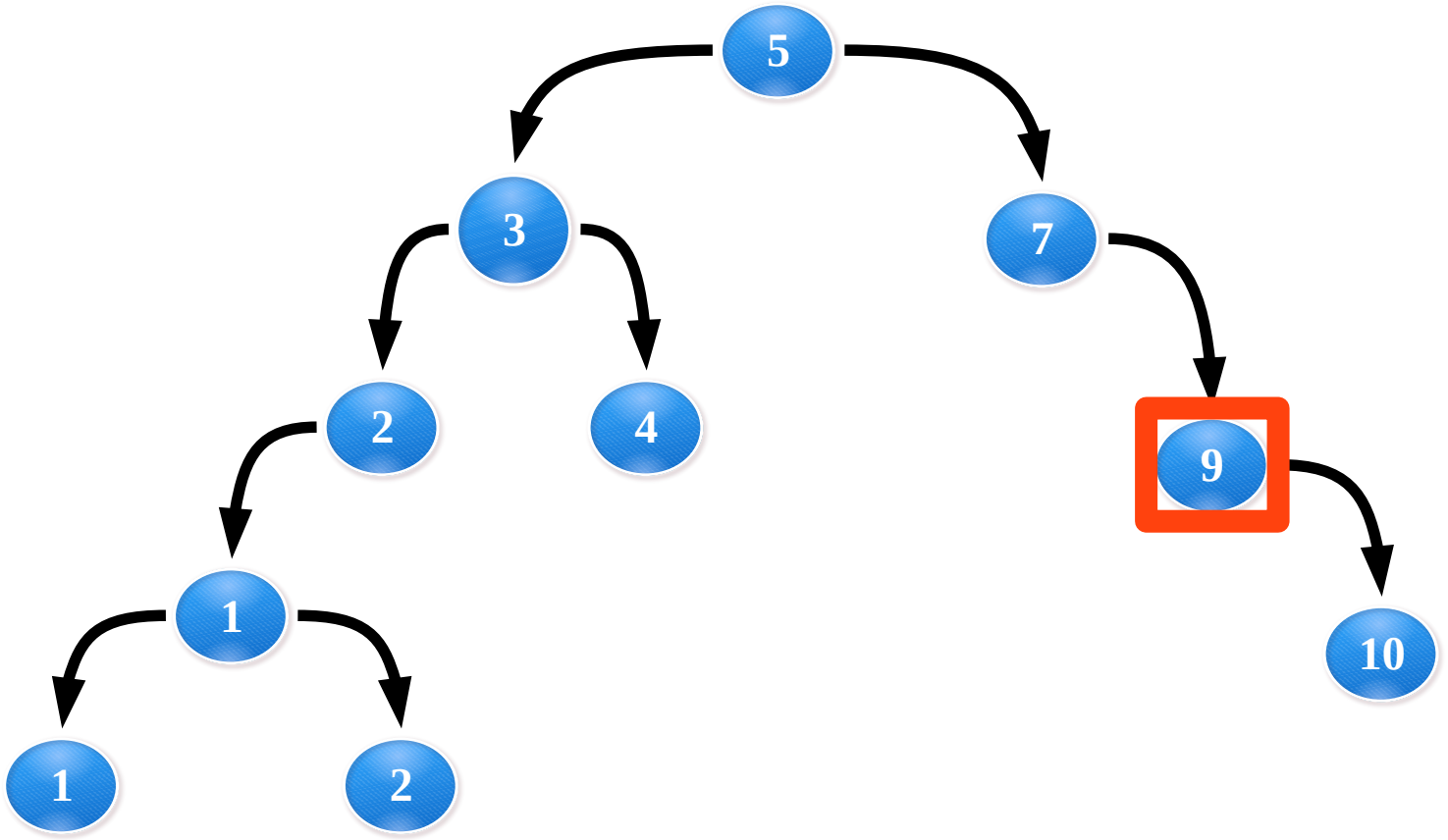
Statički alocirano stablo (DFS desno)

5	7				



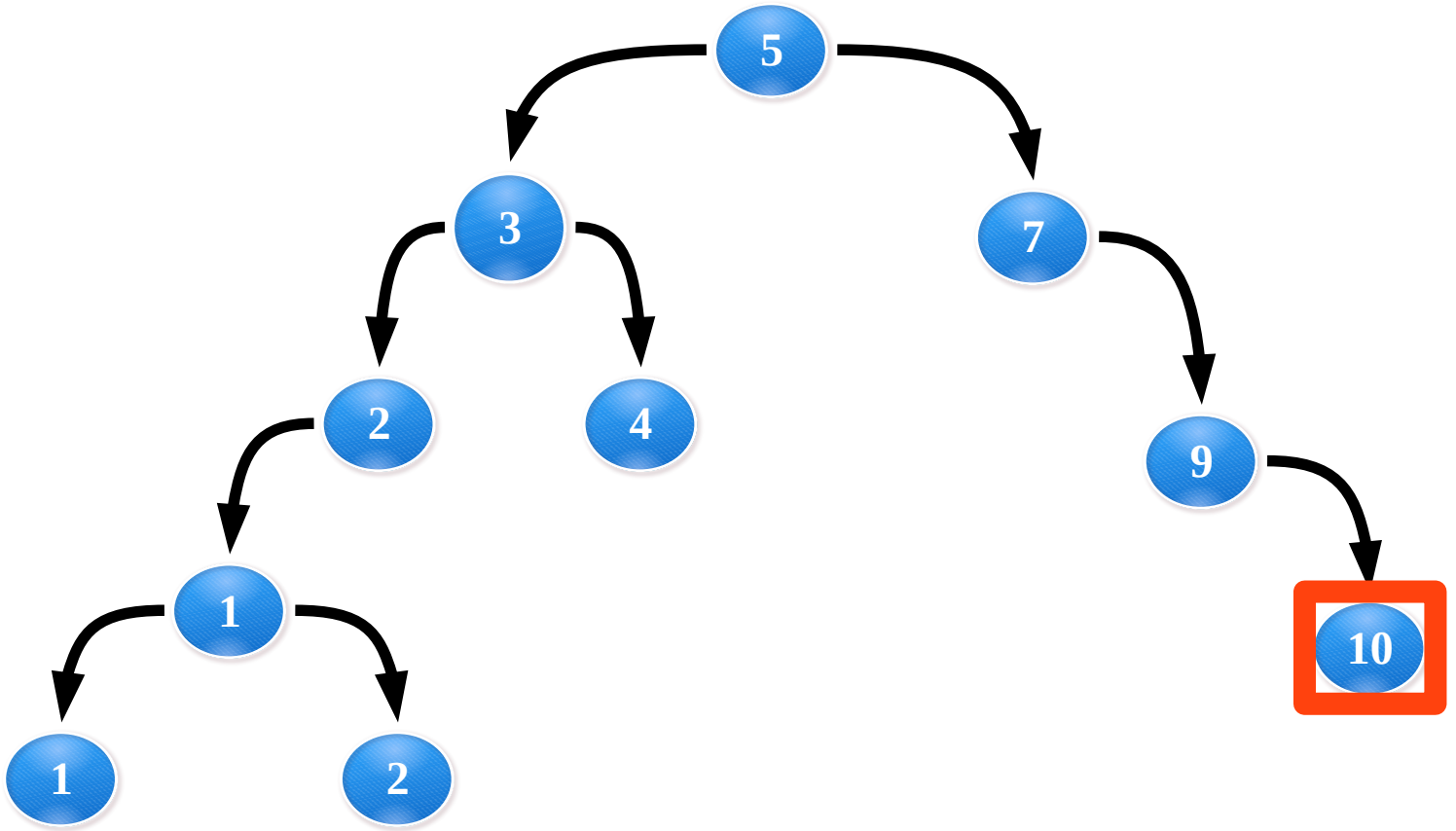
Statički alocirano stablo (DFS desno)

5	7	9				



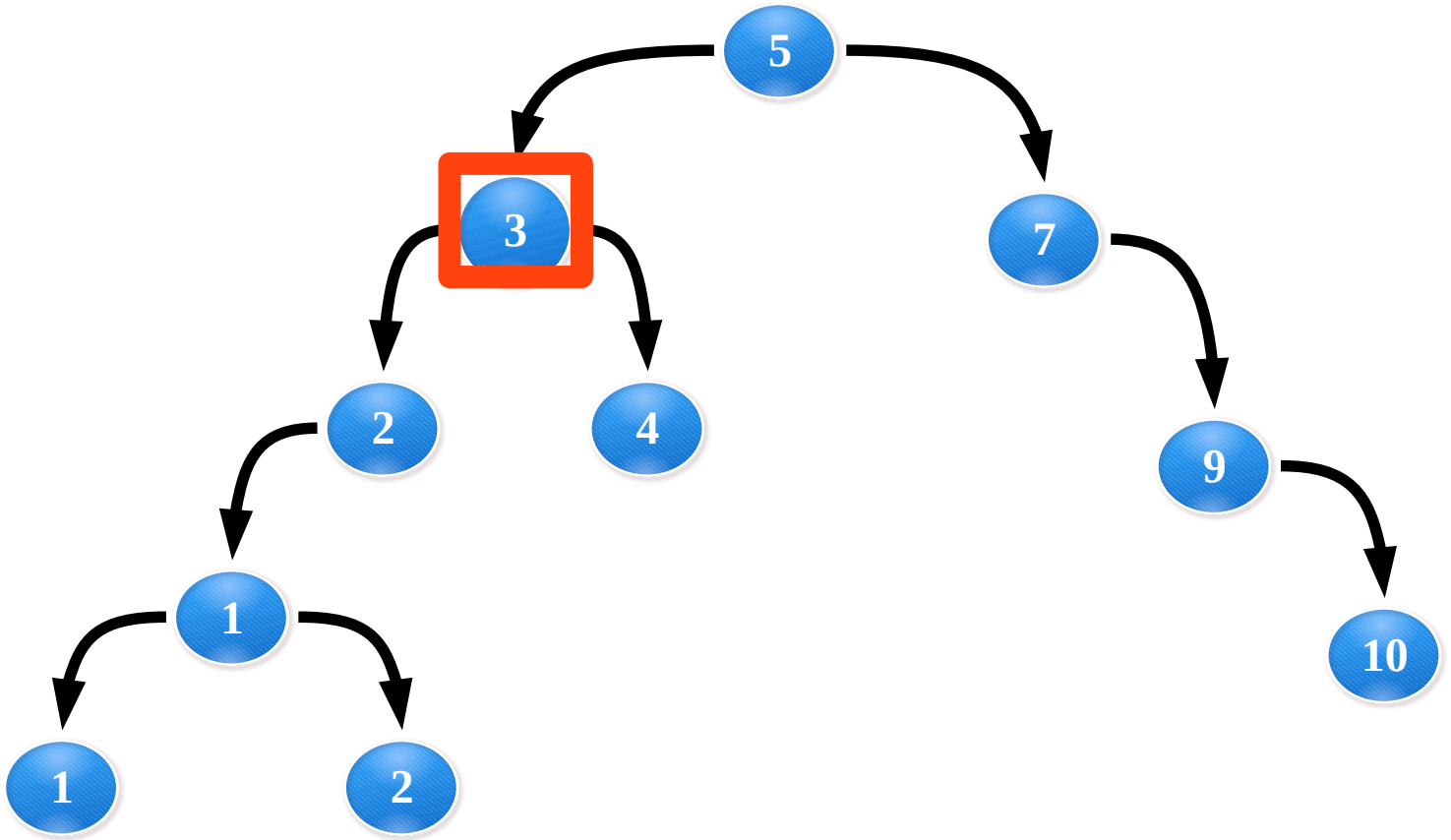
Statički alocirano stablo (DFS desno)

5	7	9	10			



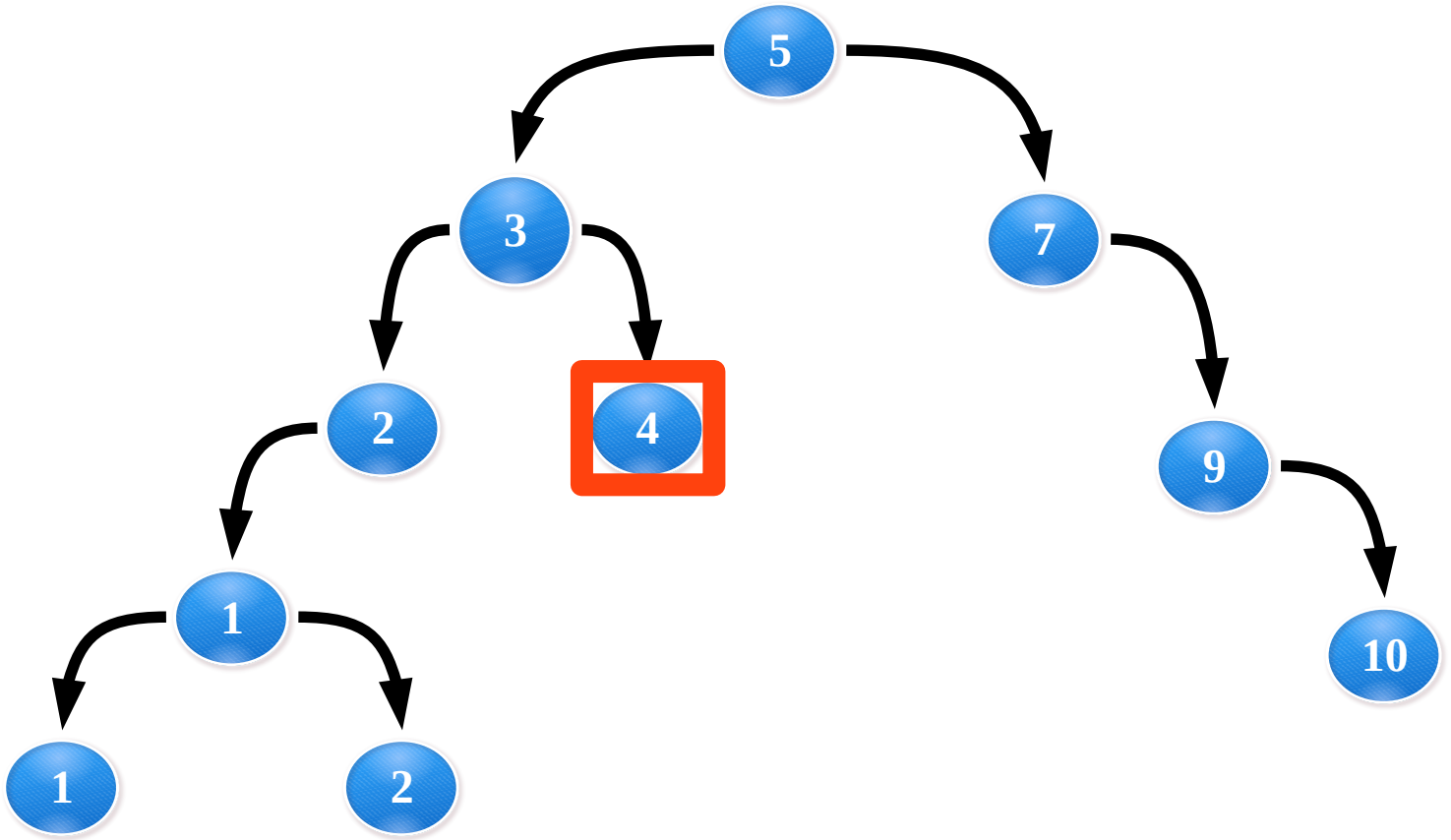
Statički alocirano stablo (DFS desno)

5	7	9	10	3			



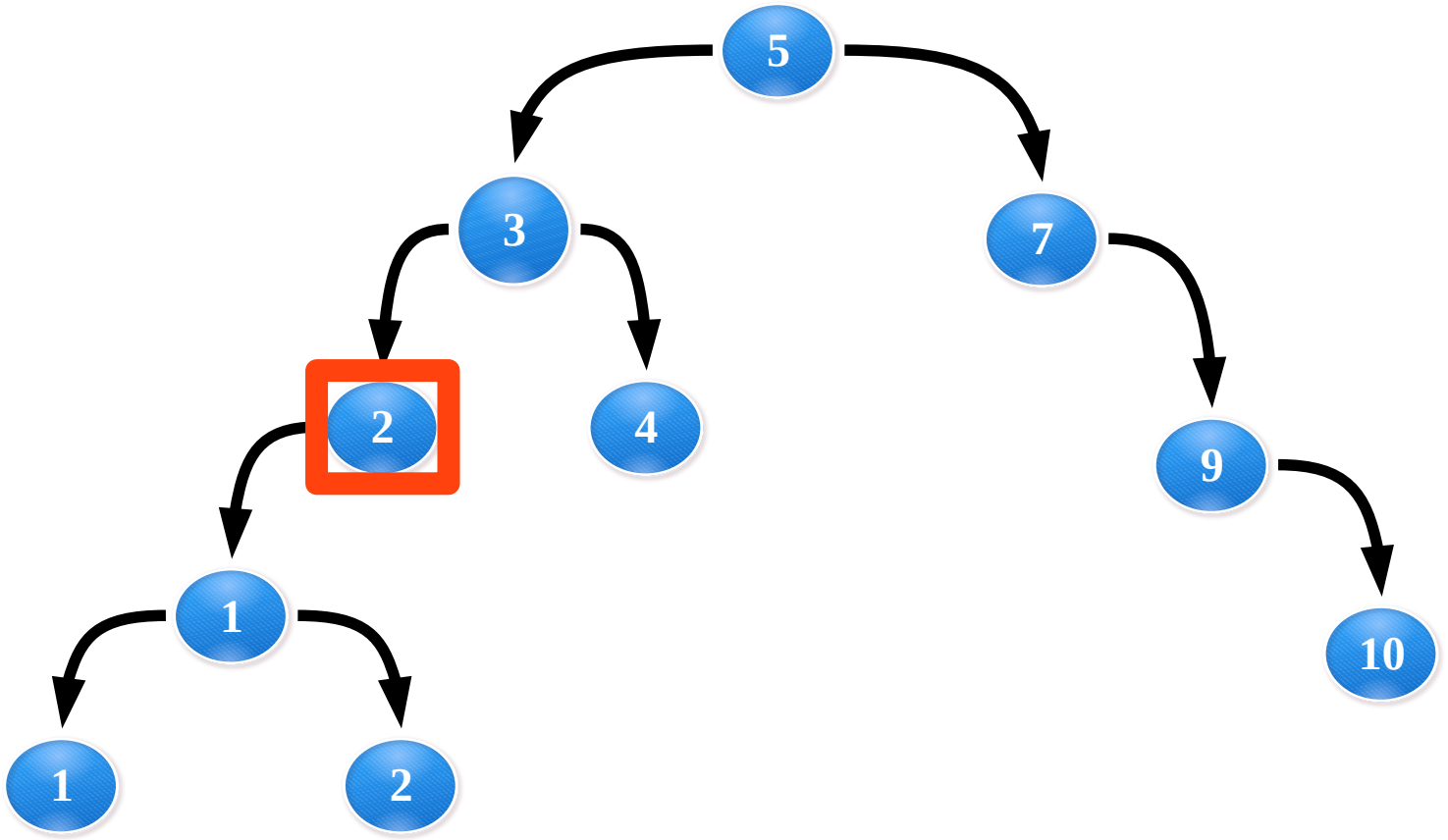
Statički alocirano stablo (DFS desno)

5		7		9	10		3		4					



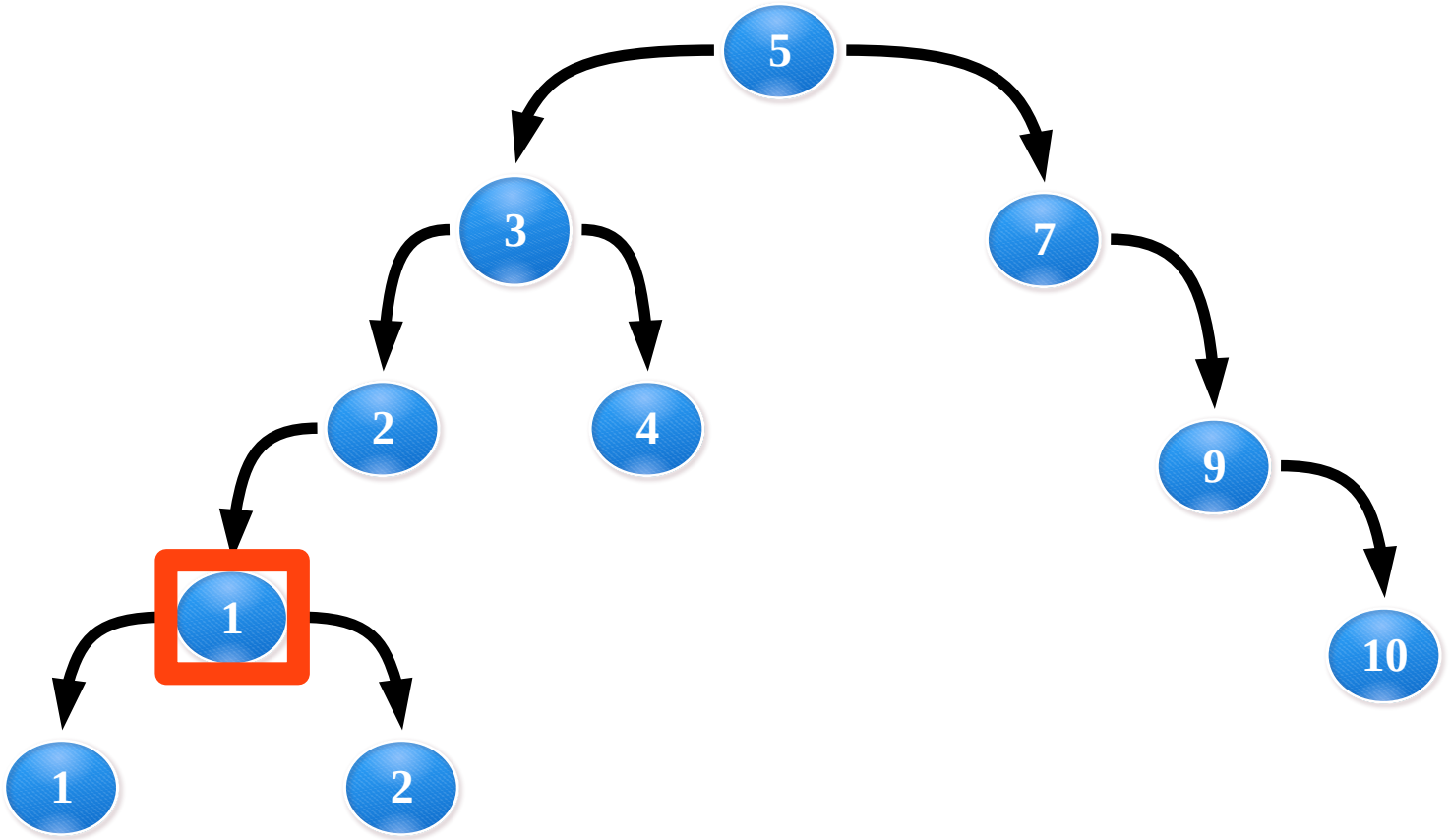
Statički alocirano stablo (DFS desno)

5	7	9	10	3	4	2		



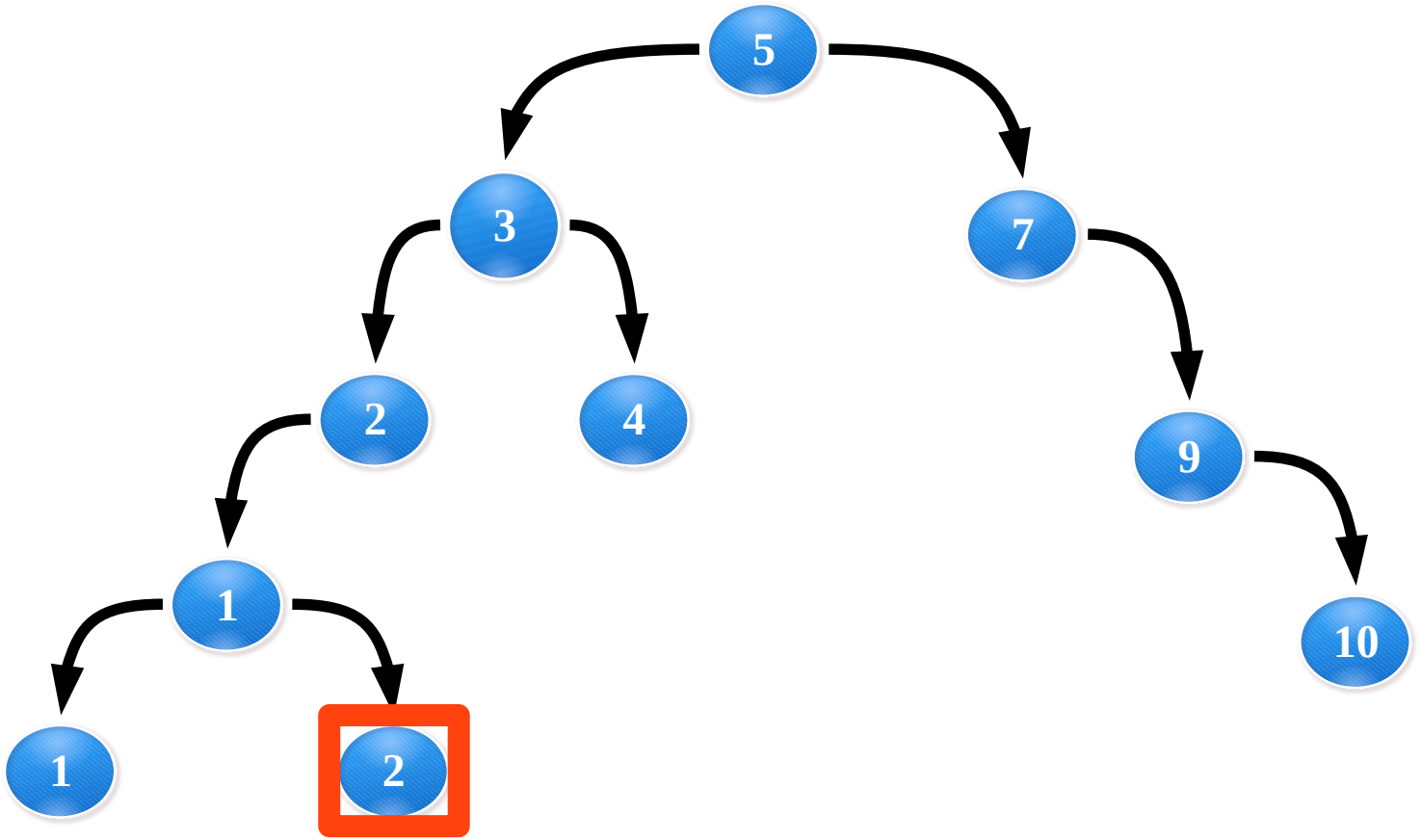
Statički alocirano stablo (DFS desno)

5	7	9	10	3	4	2	1	



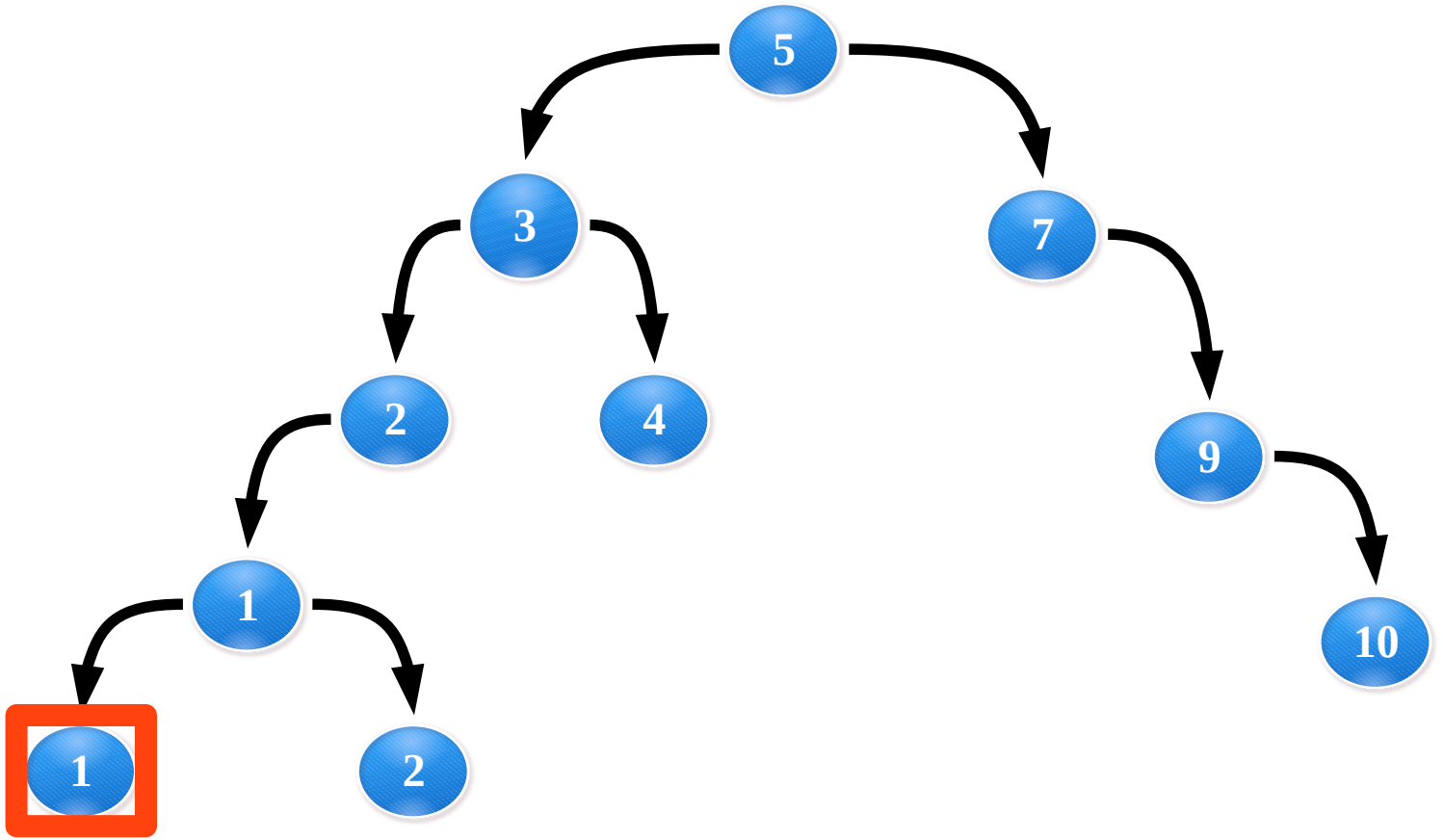
Statički alocirano stablo (DFS desno)

5	7	9	10	3	4	2	1	2	



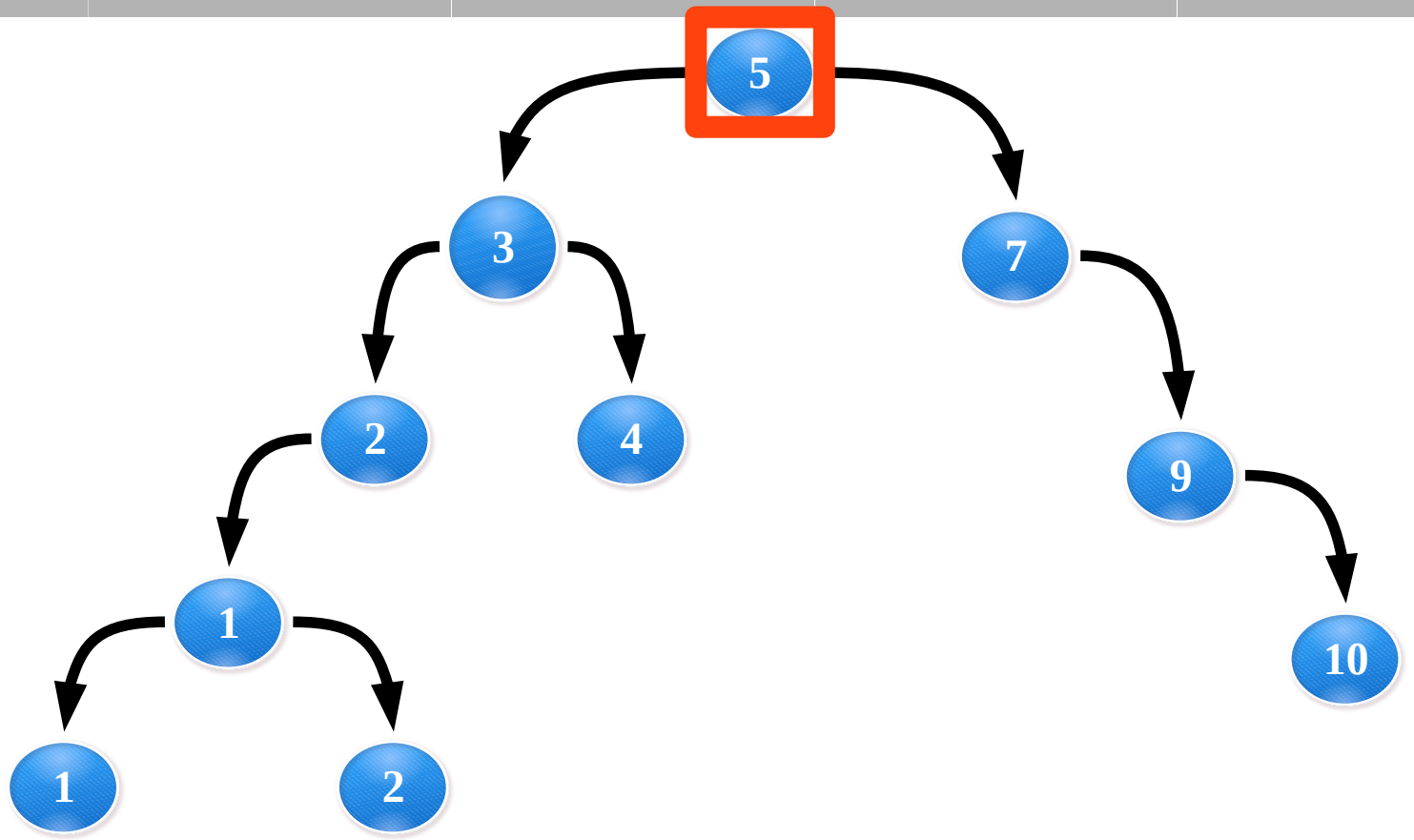
Statički alocirano stablo (DFS desno)

5	7	9	10	3	4	2	1	2	1



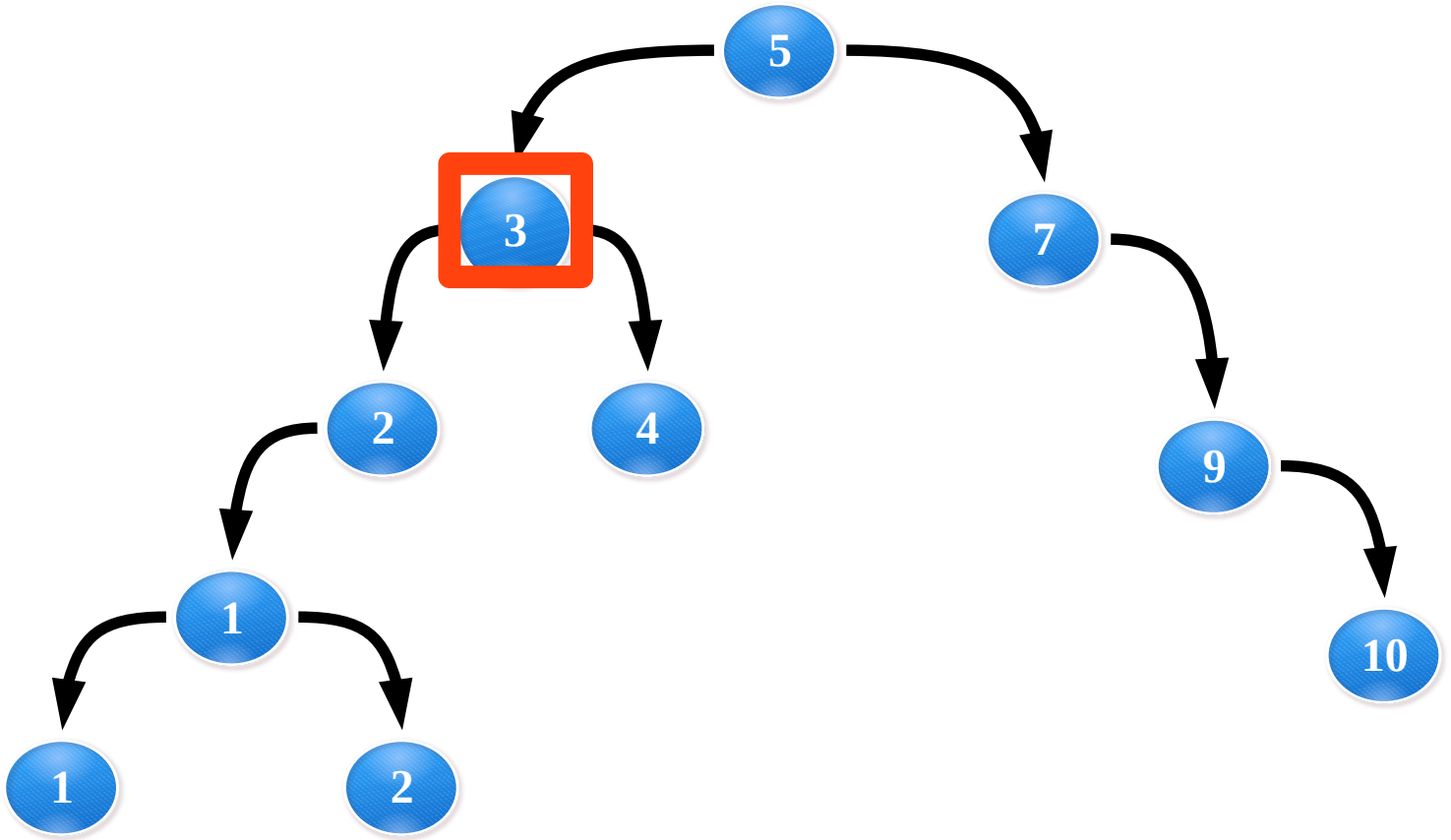
Statički alocirano stablo (BFS)

5					



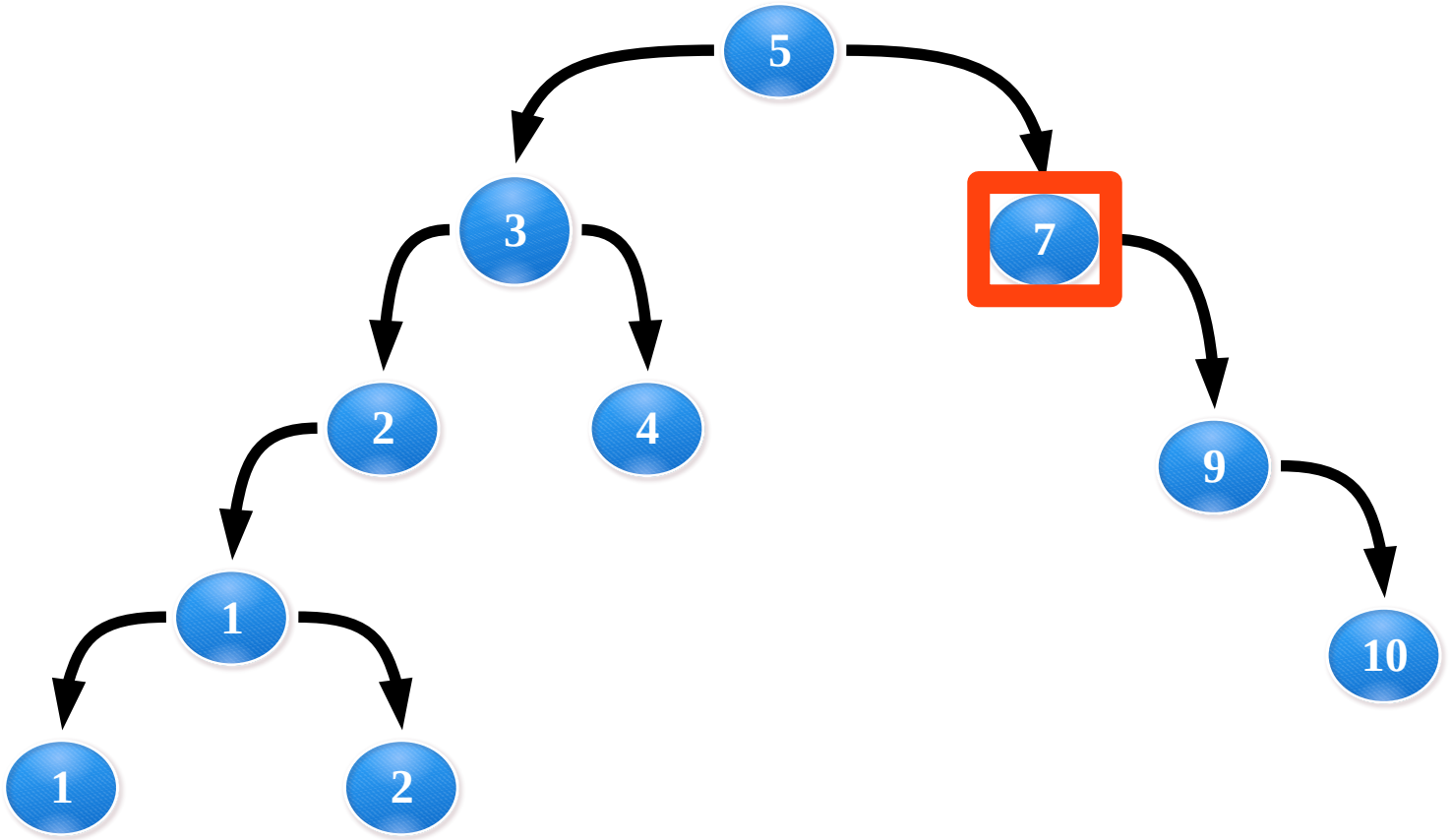
Statički alocirano stablo (BFS)

5	3				



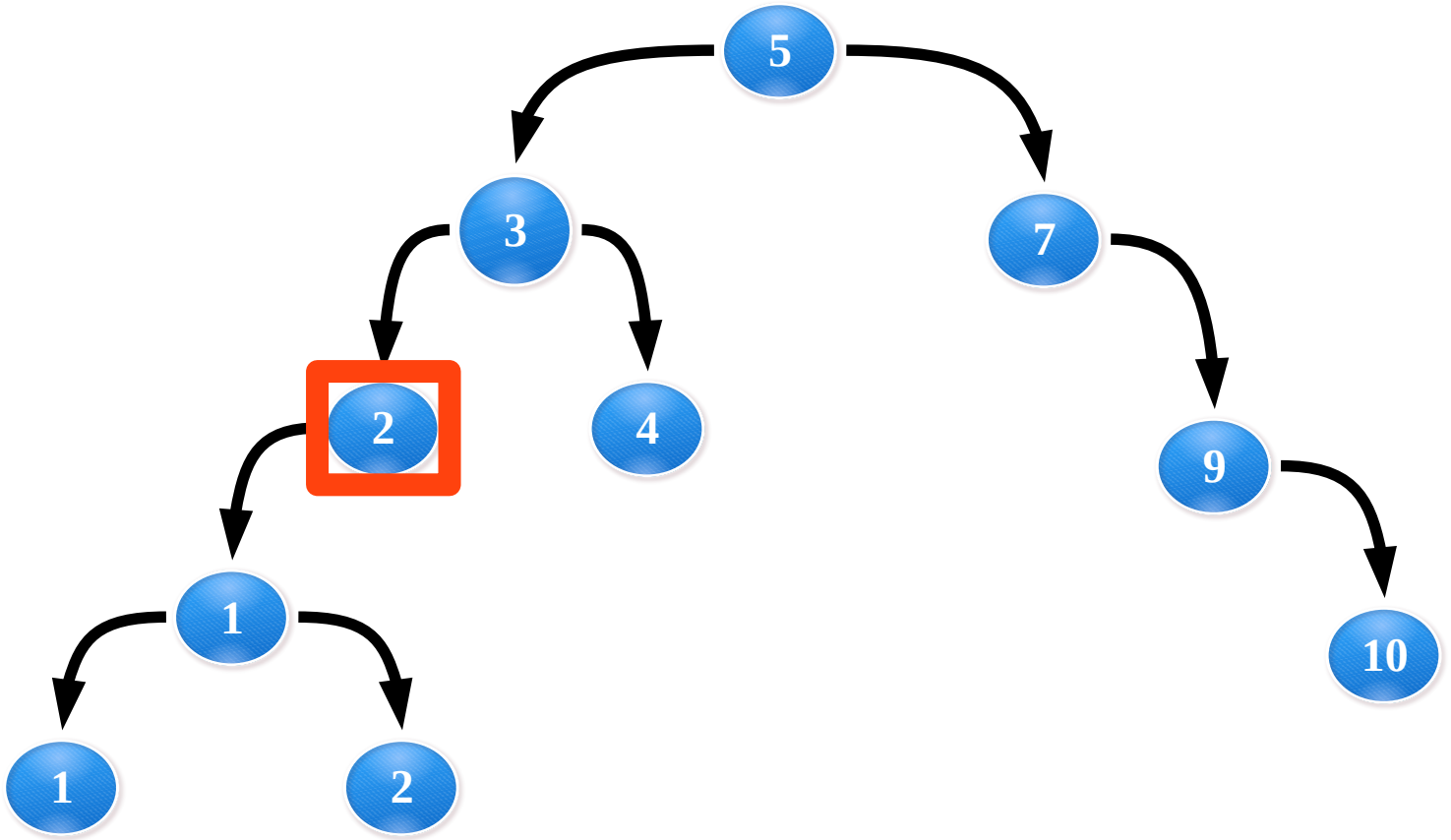
Statički alocirano stablo (BFS)

5	3	7				



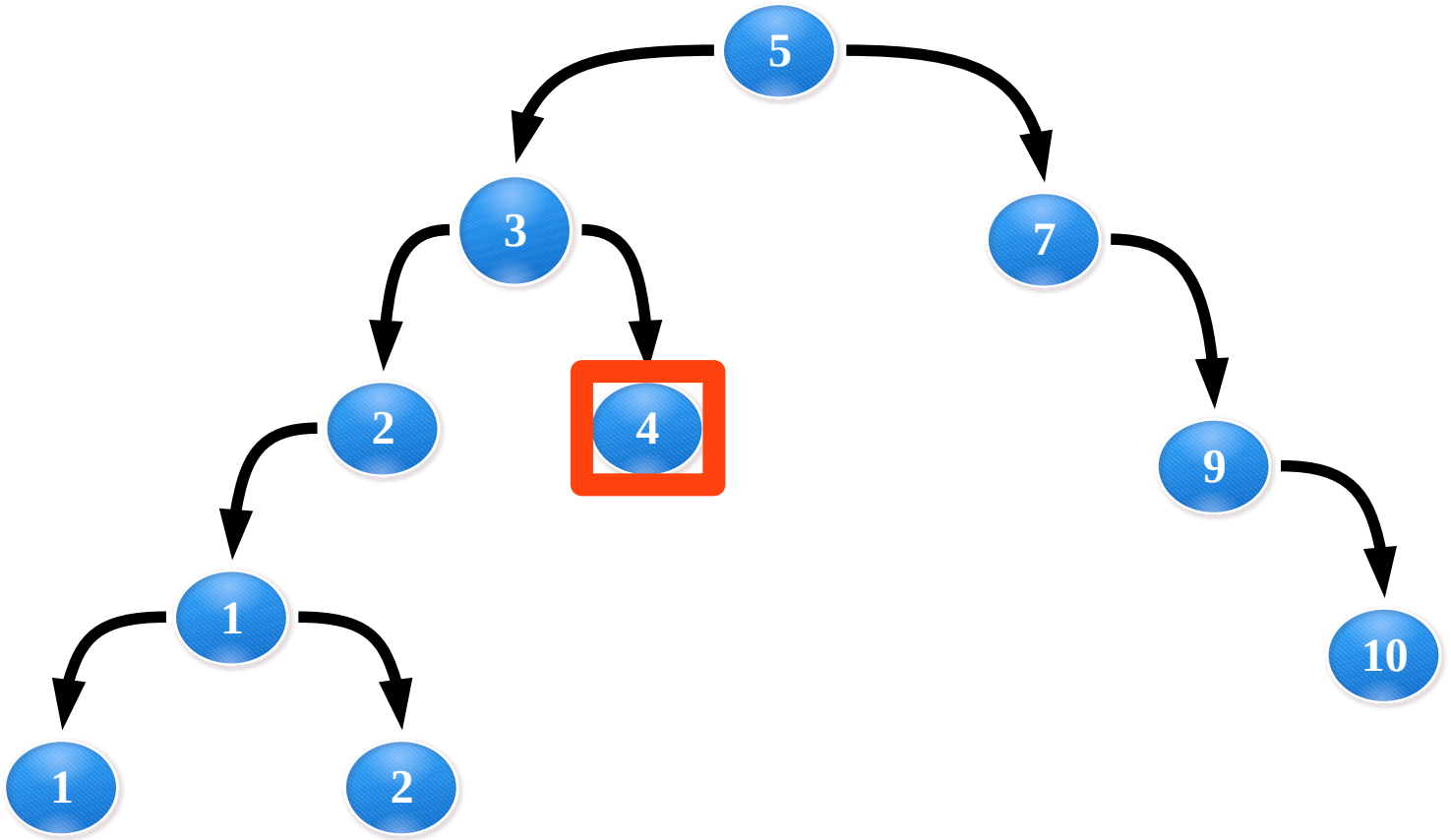
Statički alocirano stablo (BFS)

5	3	7	2			



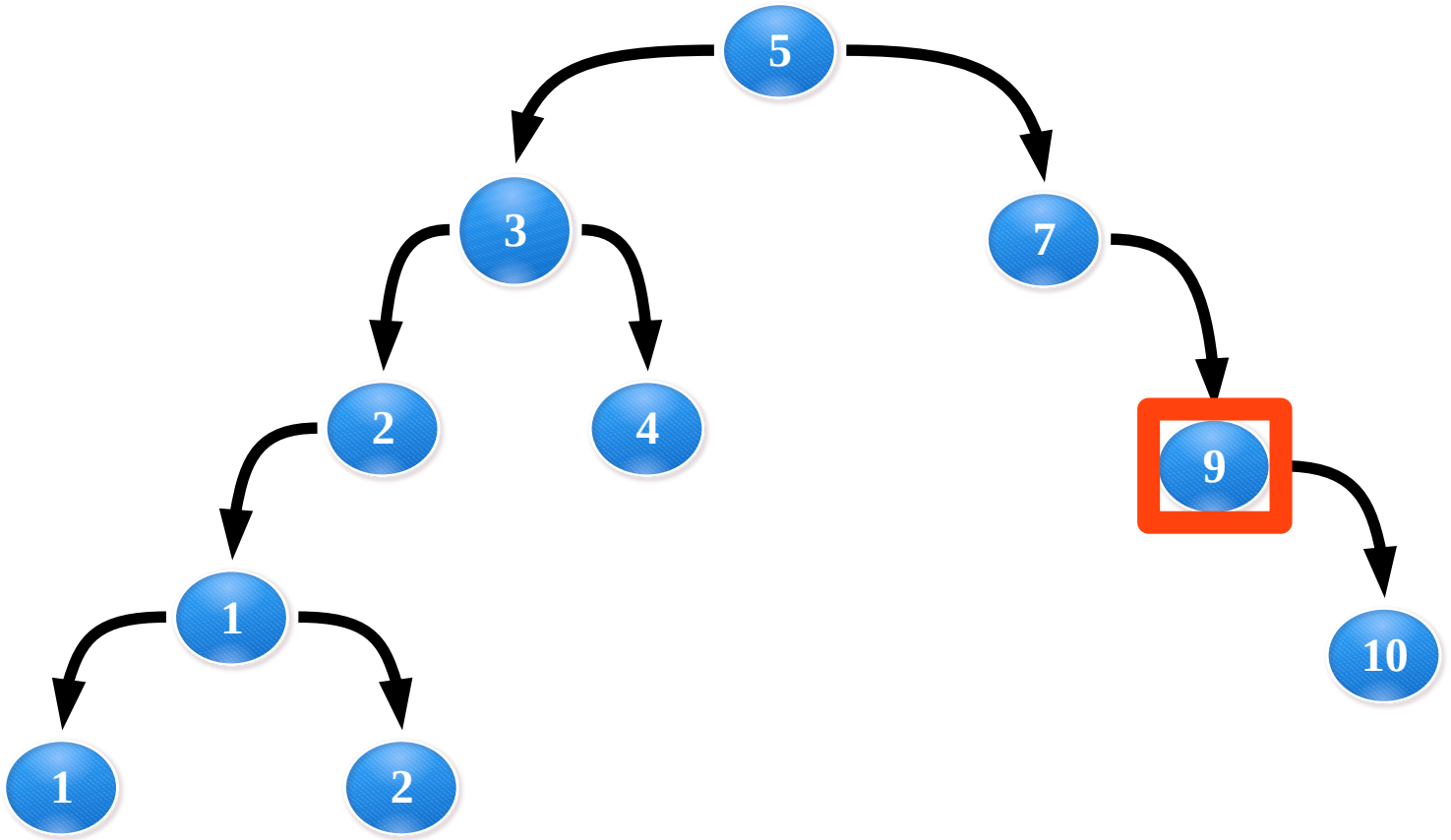
Statički alocirano stablo (BFS)

5	3	7	2	4			



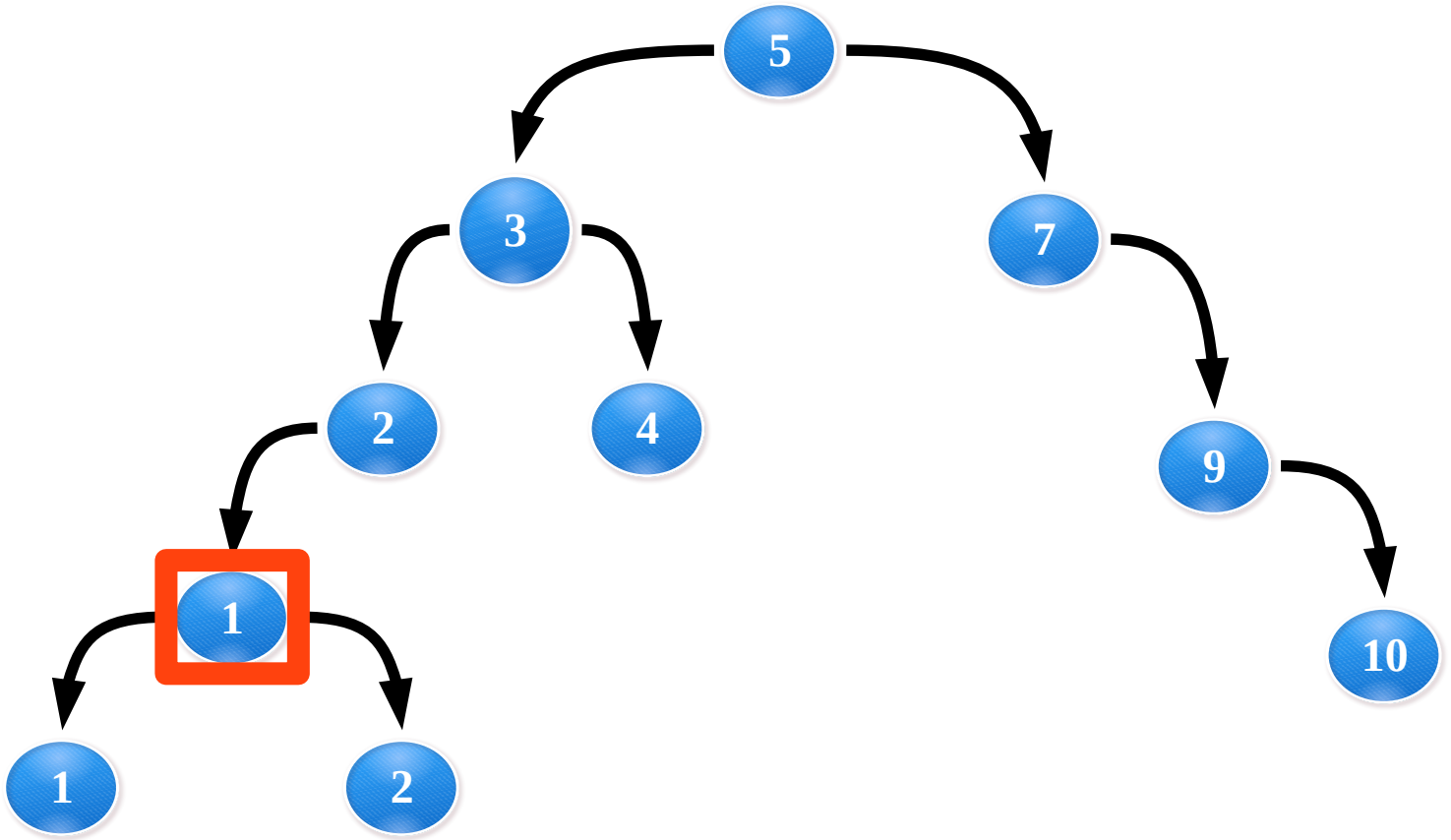
Statički alocirano stablo (BFS)

5	3	7	2	4	9		



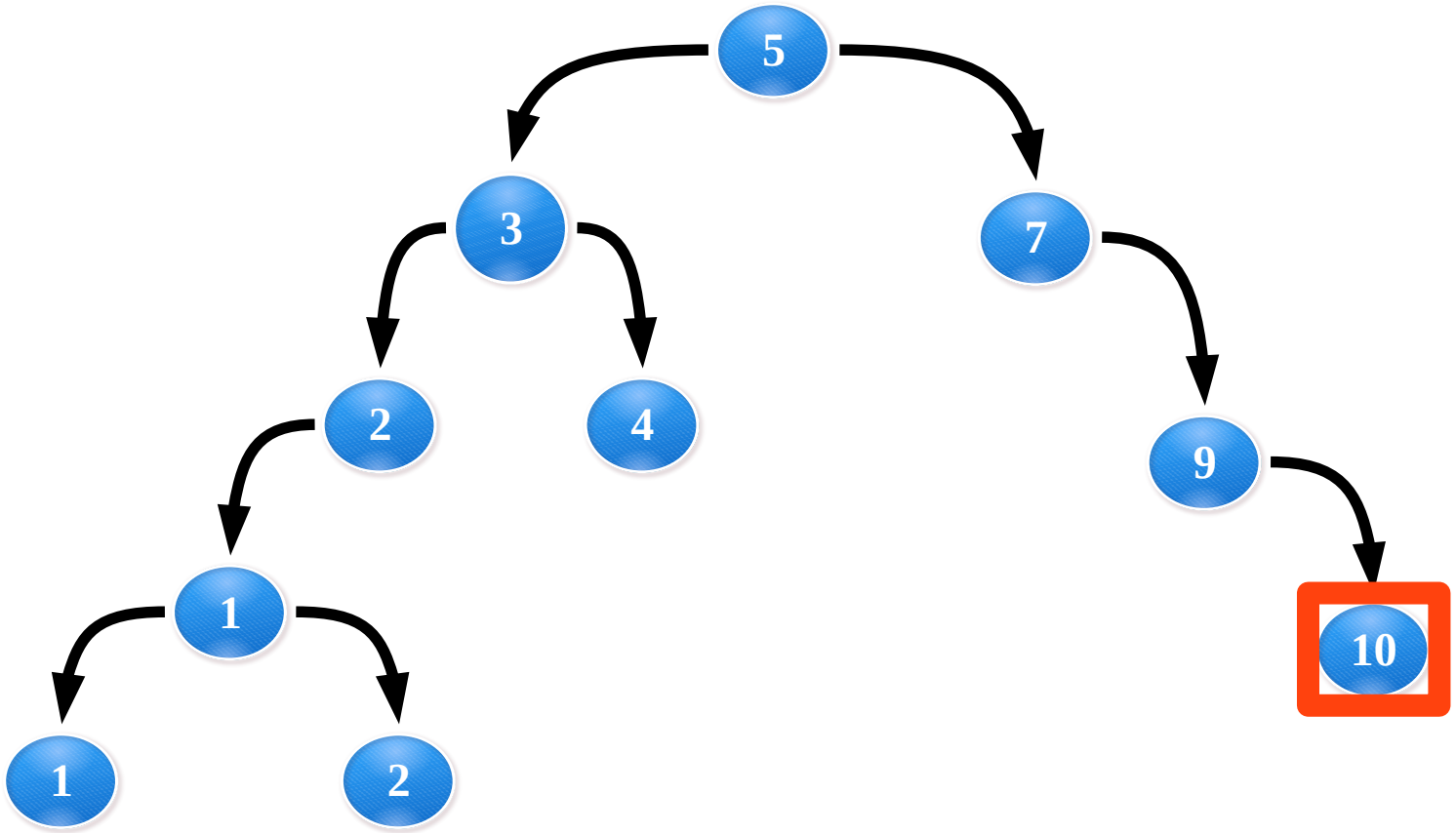
Statički alocirano stablo (BFS)

5	3	7	2	4	9	1		



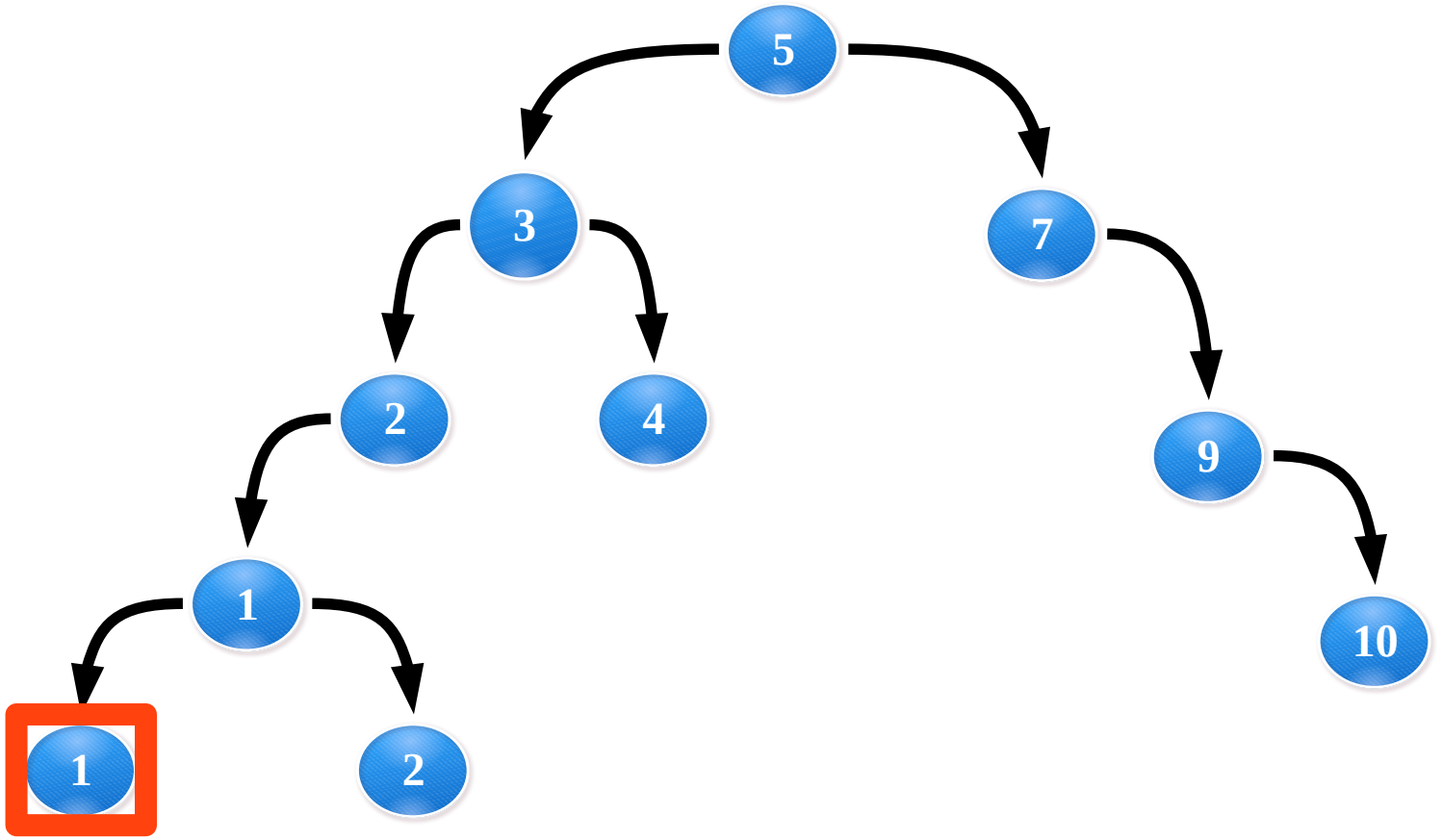
Statički alocirano stablo (BFS)

5	3	7	2	4	9	1	10	



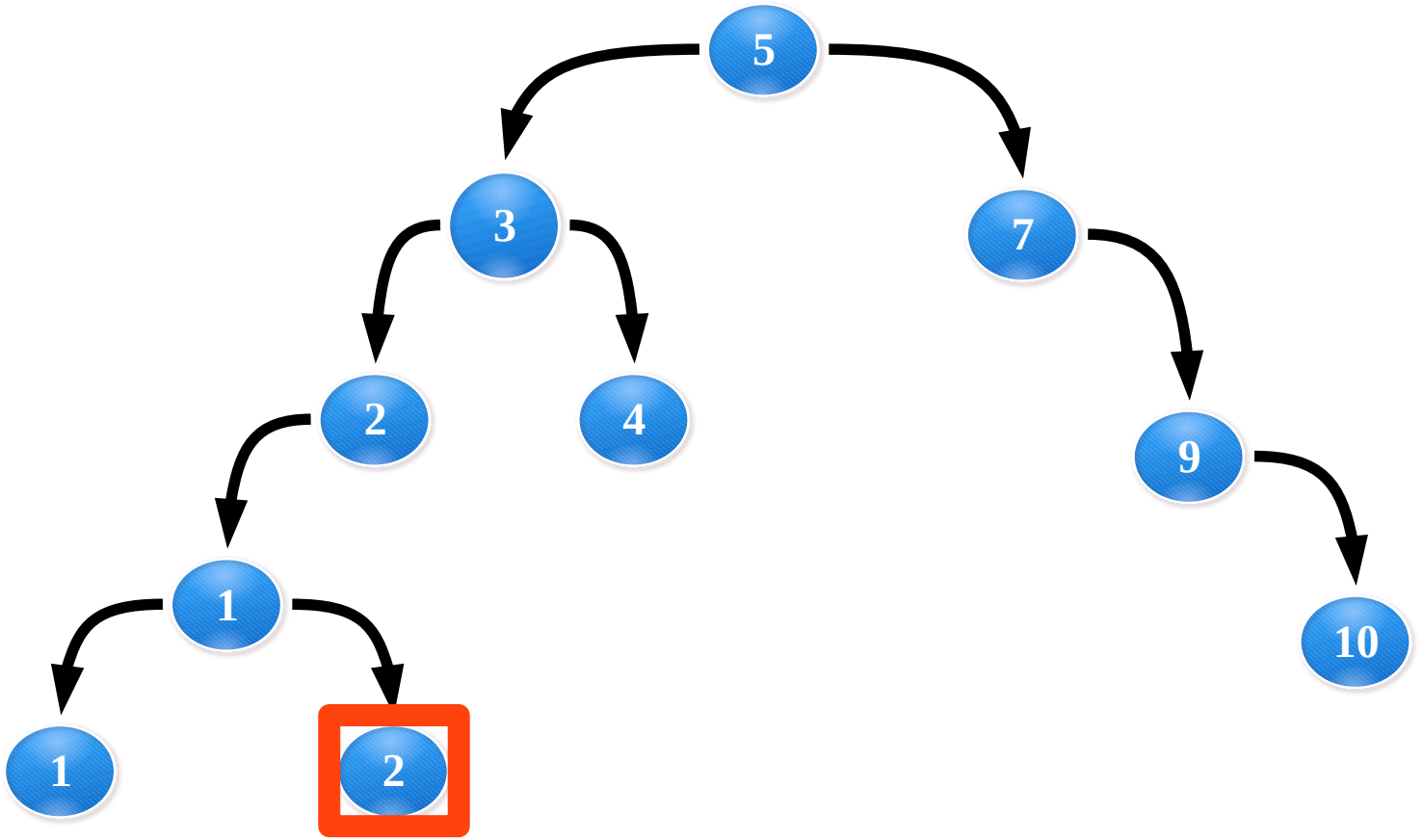
Statički alocirano stablo (BFS)

5	3	7	2	4	9	1	10	1	



Statički alocirano stablo (BFS)

5	3	7	2	4	9	1	10	1	2

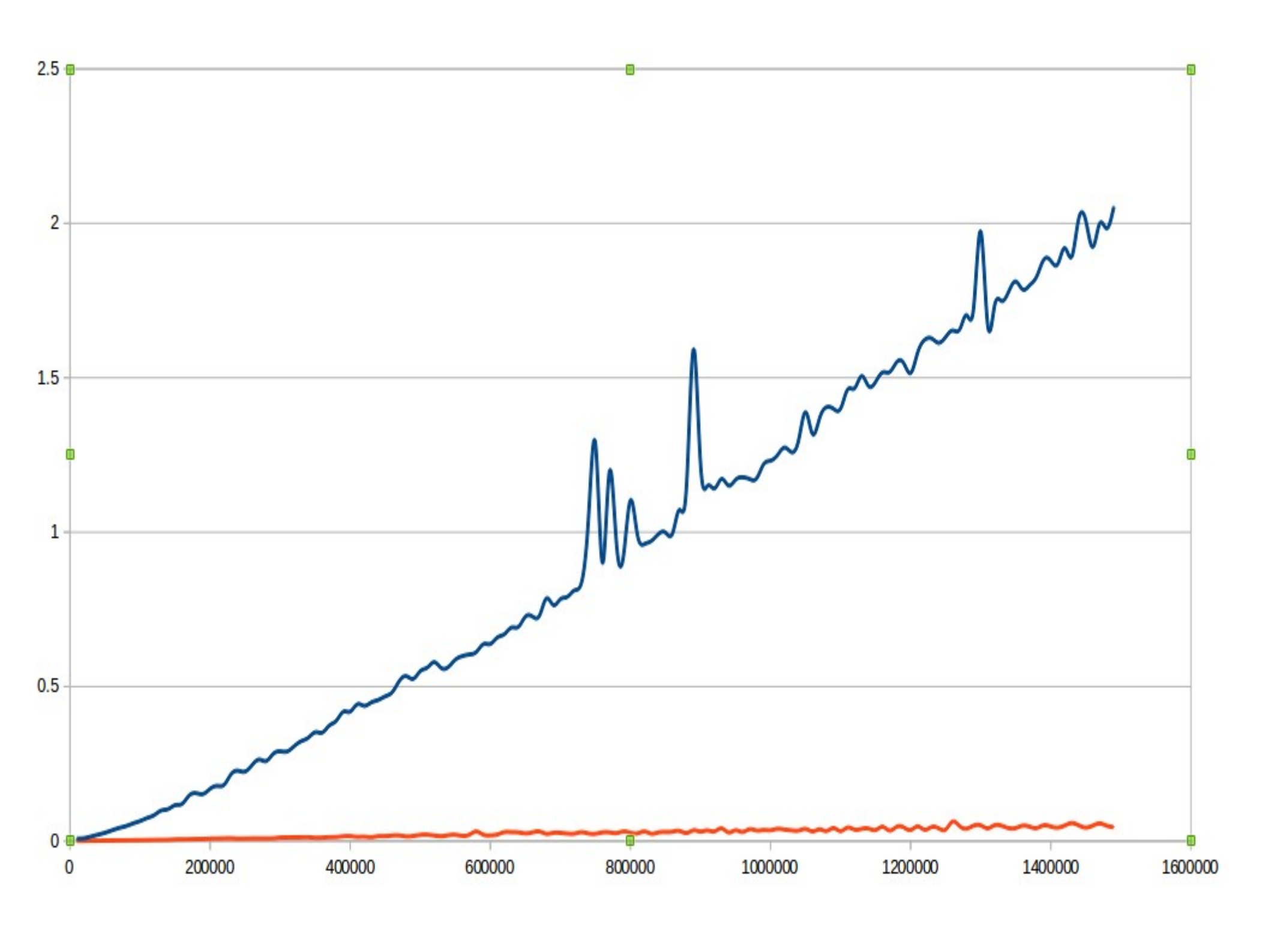


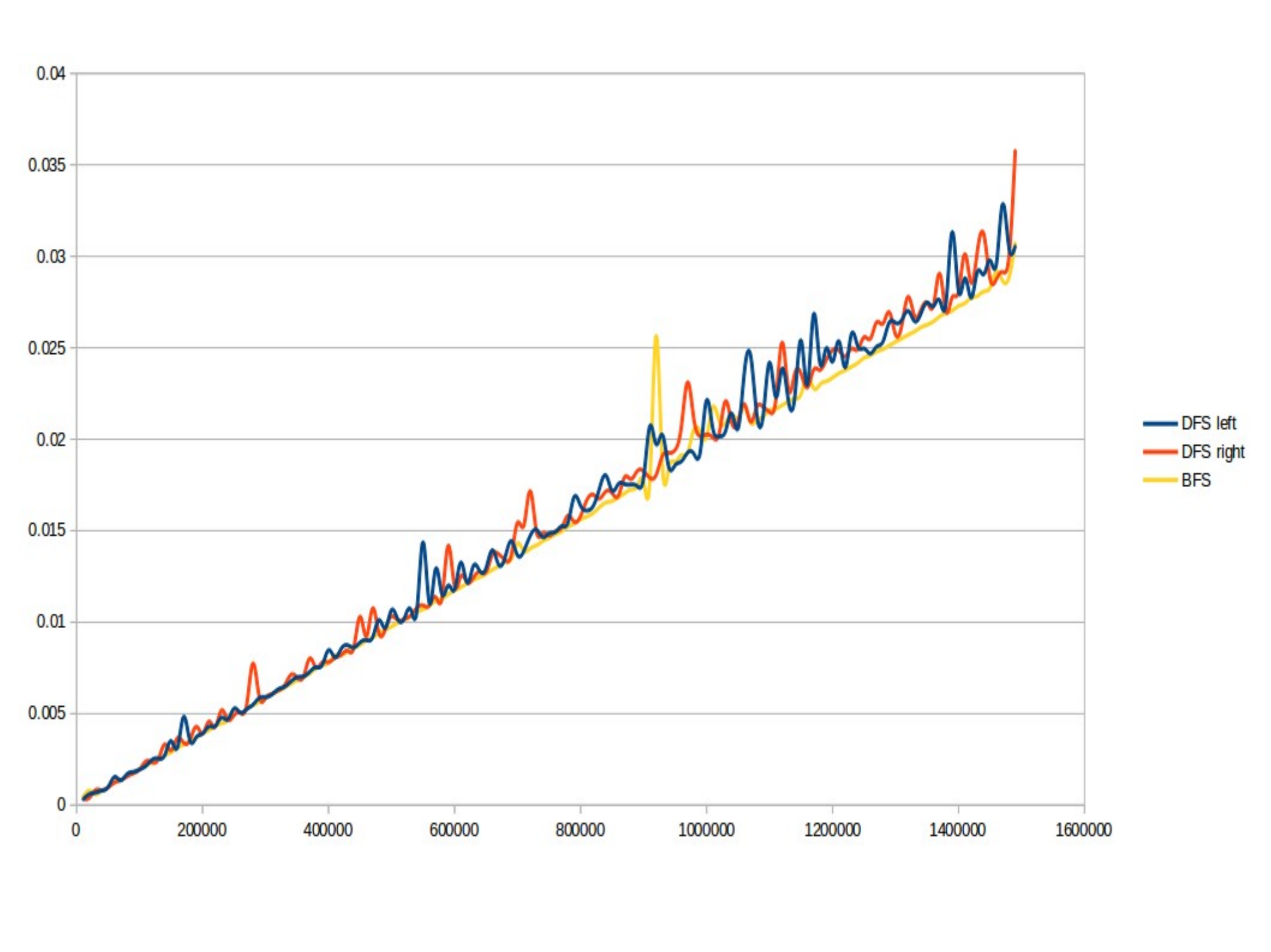
Statički alocirano stablo

```
 81 template <class T>
 82 int CacheOptimizedStaticTree<T>::createCacheOptimizedStaticTree(
 83                                                 BinaryNode<T>* node, int current) {
 84   if (node == NULL) return current-1;
 85   BinaryNode<T> current_node(node->data);
 86   tree.push_back(current_node);
 87   int left = createCacheOptimizedStaticTree(node->left, current+1);
 88   if (left == current) {
 89     current_node.left = NULL;
 90   } else {
 91     current_node.left = &tree[left];
 92   }
 93   int right = createCacheOptimizedStaticTree(node->right, left+1);
 94   if (right == left) {
 95     current_node.right = NULL;
 96   } else {
 97     current_node.right = &tree[right];
 98   }
 99   return current;
100 }
```

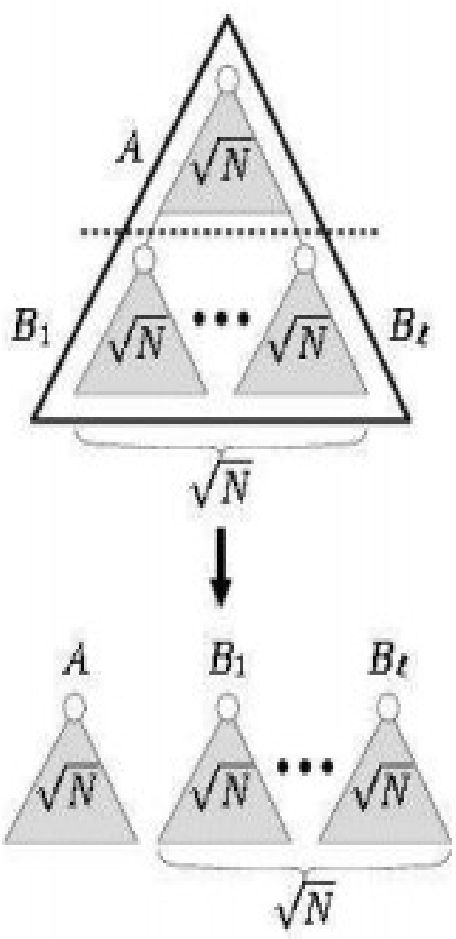
Search

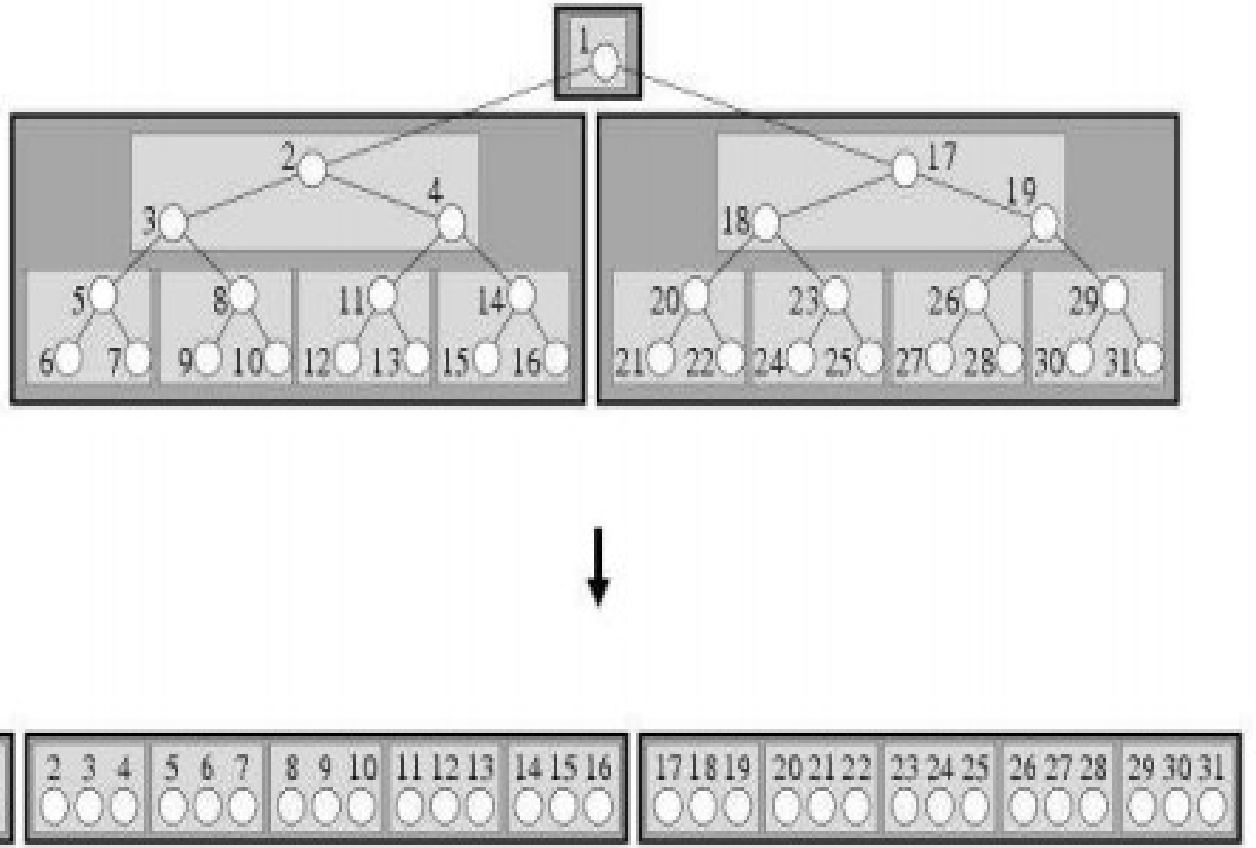
```
template <class T>
bool CacheOptimizedStaticTree<T>::contains(T elem) const{
  if (isEmpty()) return false;
  BinaryNode<T> node = tree[0];

  while(true) {
    if(node.data == elem) return true;
    else if (node.data < elem) {
      if (node.left == NULL) return false;
      node = *node.left;
    }
    else {
      if (node.right == NULL) return false;
      node = *node.right;
    }
  }
```


2.5
2
1.5
1
0.5
0
0
200000
400000
600000
800000
1000000
1200000
1400000
1600000

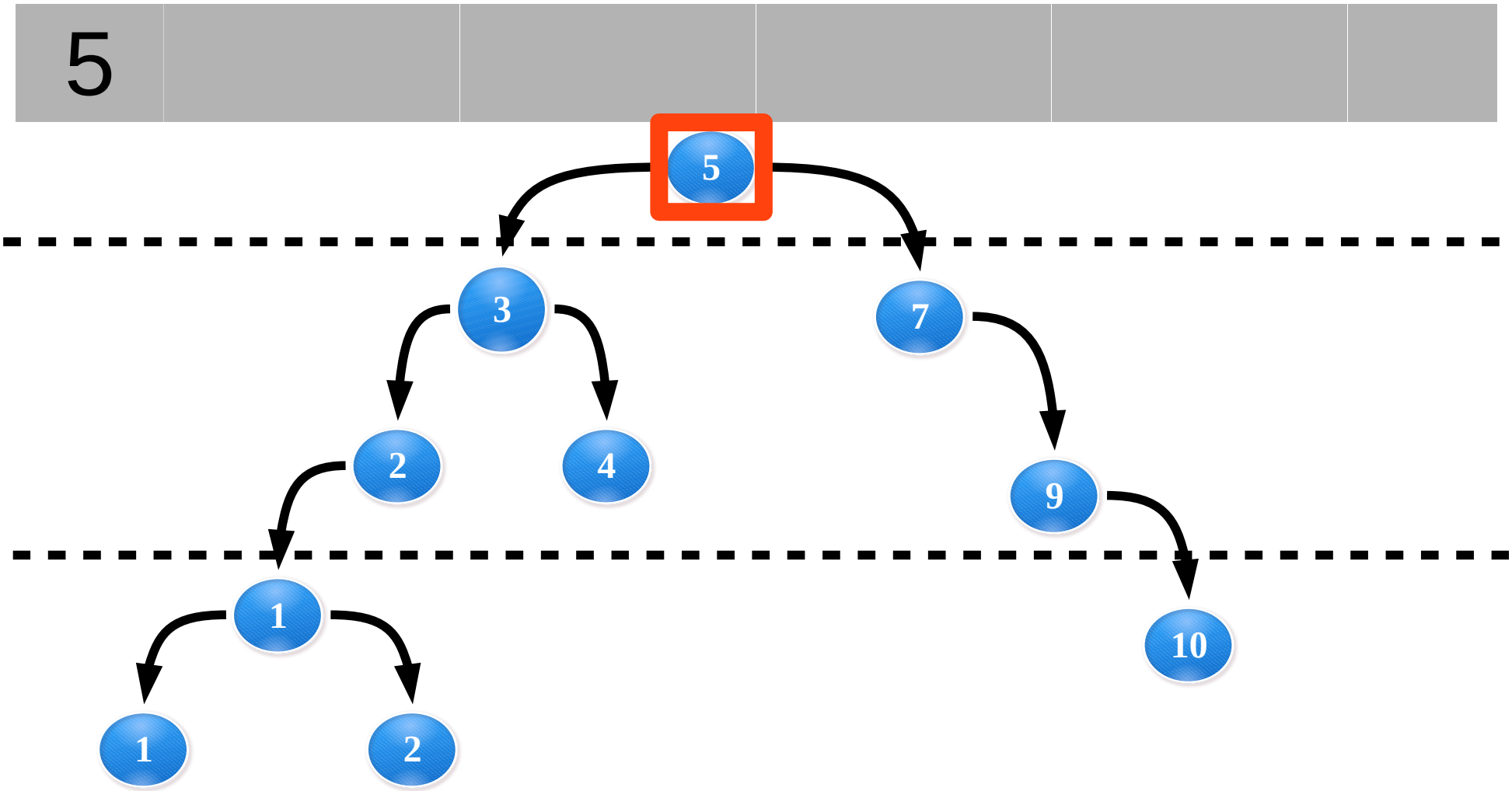

0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
200000
400000
600000
800000
1000000
1200000
1400000
1600000
DFS left
DFS right
BFS

van Emde Boas raspored



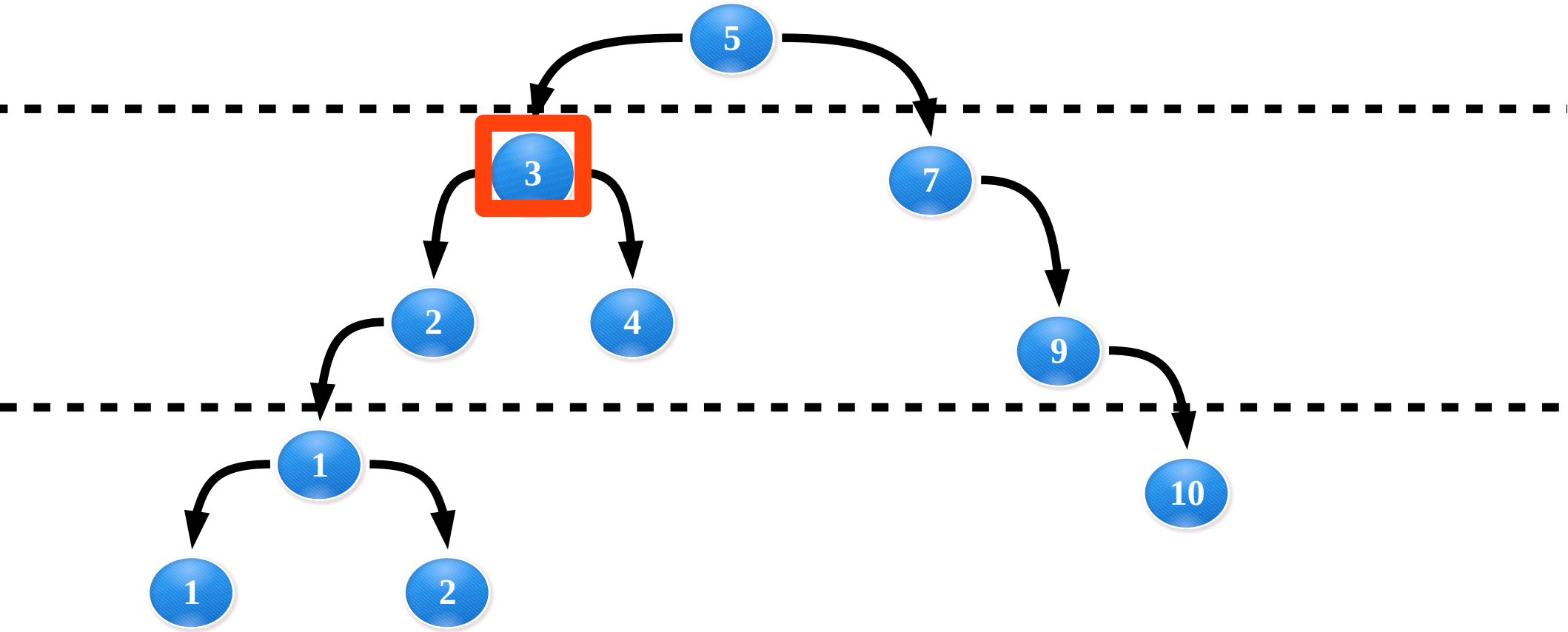


Statički alocirano stablo (Emde)



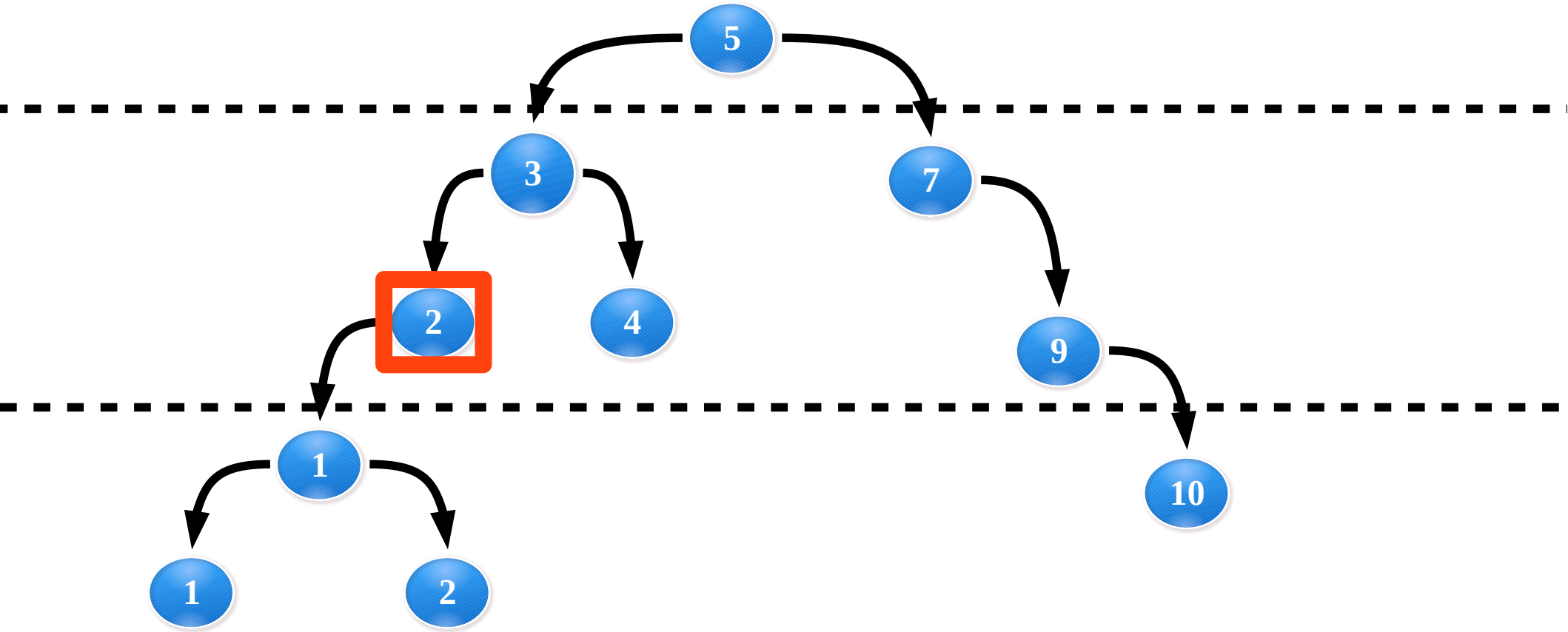
Statički alocirano stablo (Emde)

5	3				



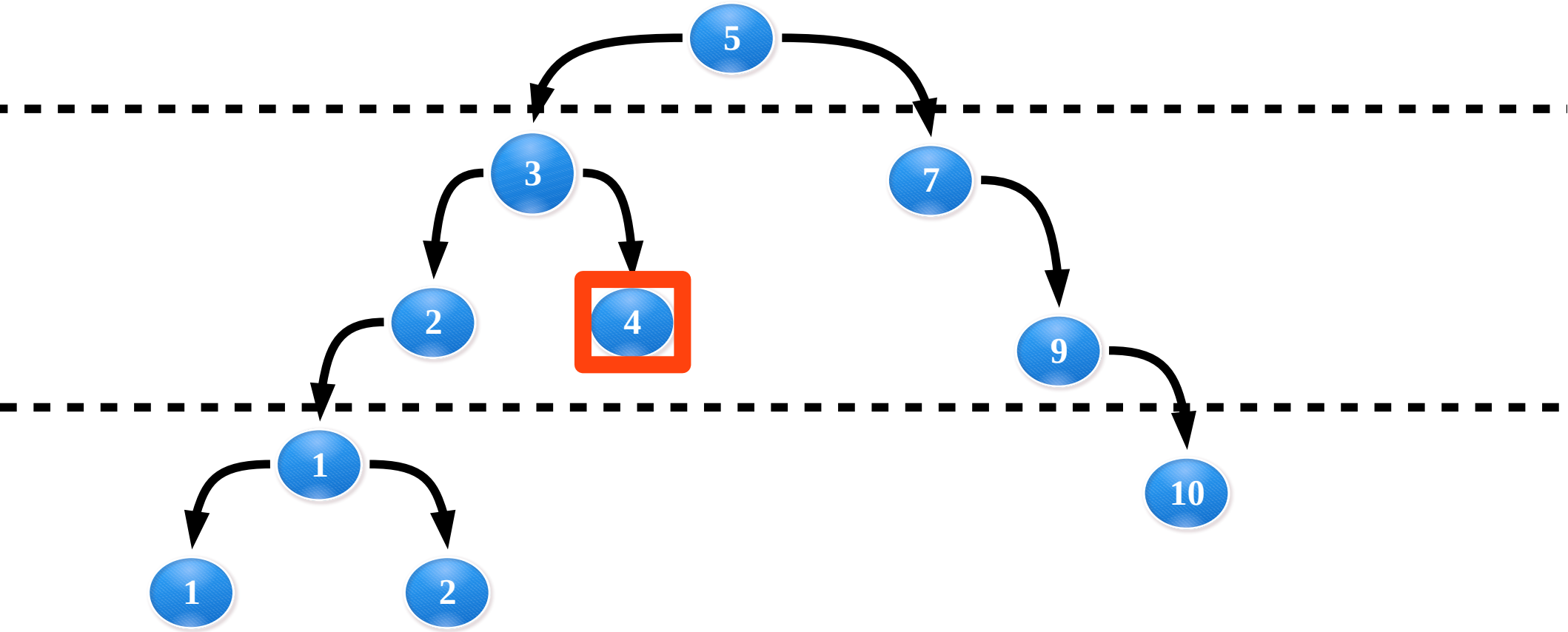
Statički alocirano stablo (Emde)

5	3	2				



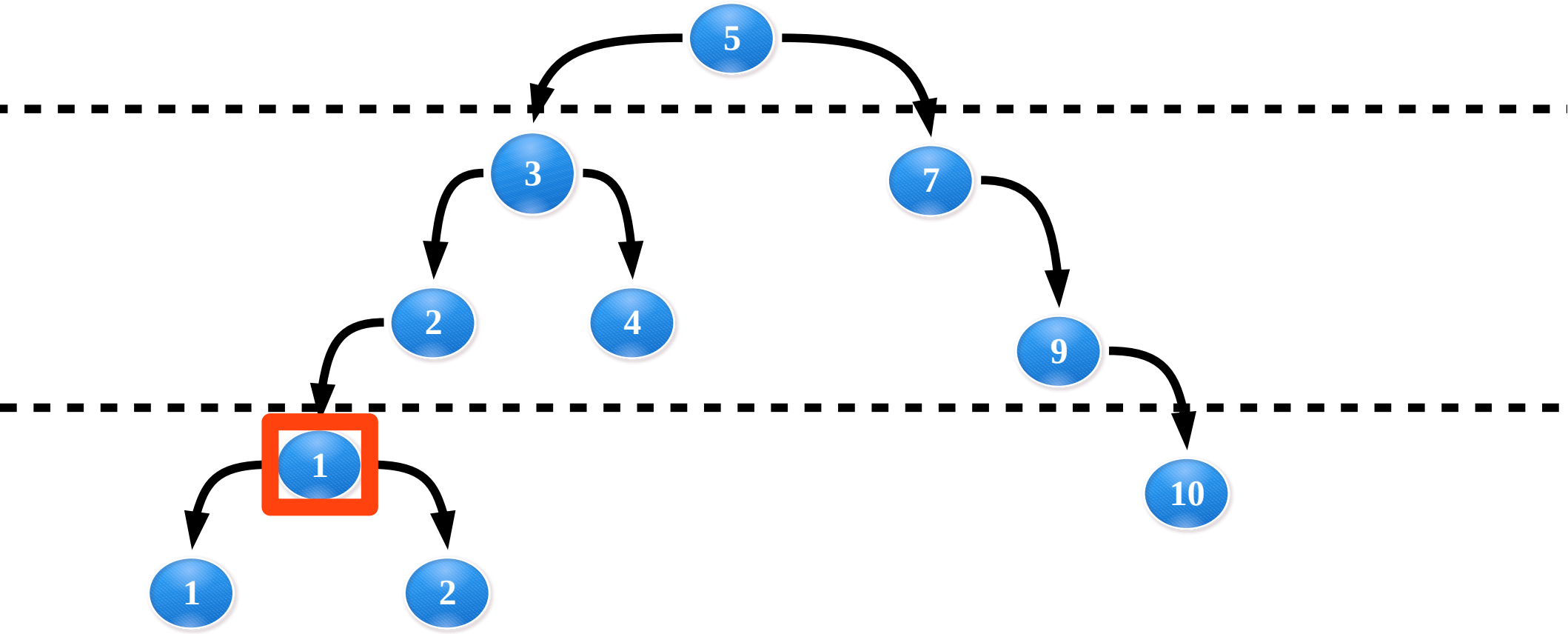
Statički alocirano stablo (Emde)

5	3	2	4				



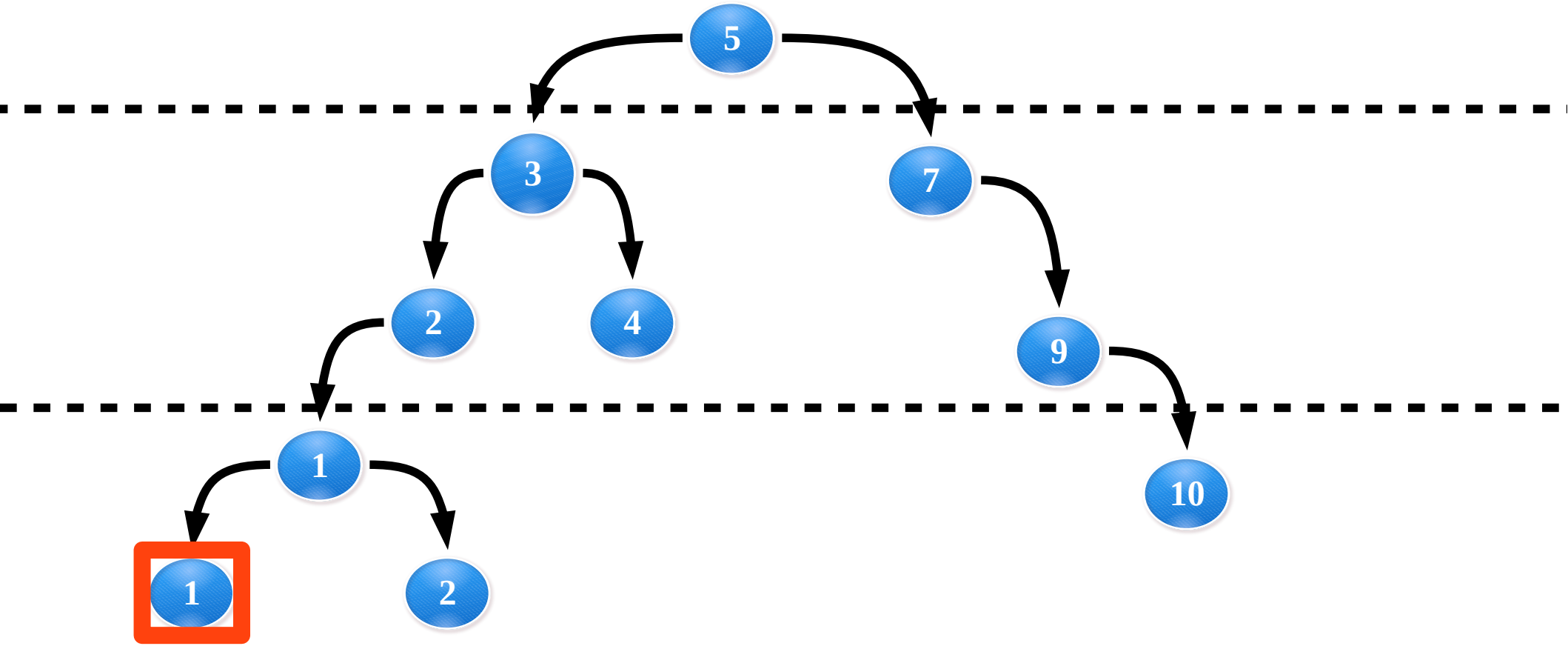
Statički alocirano stablo (Emde)

5	3	2	4	1			



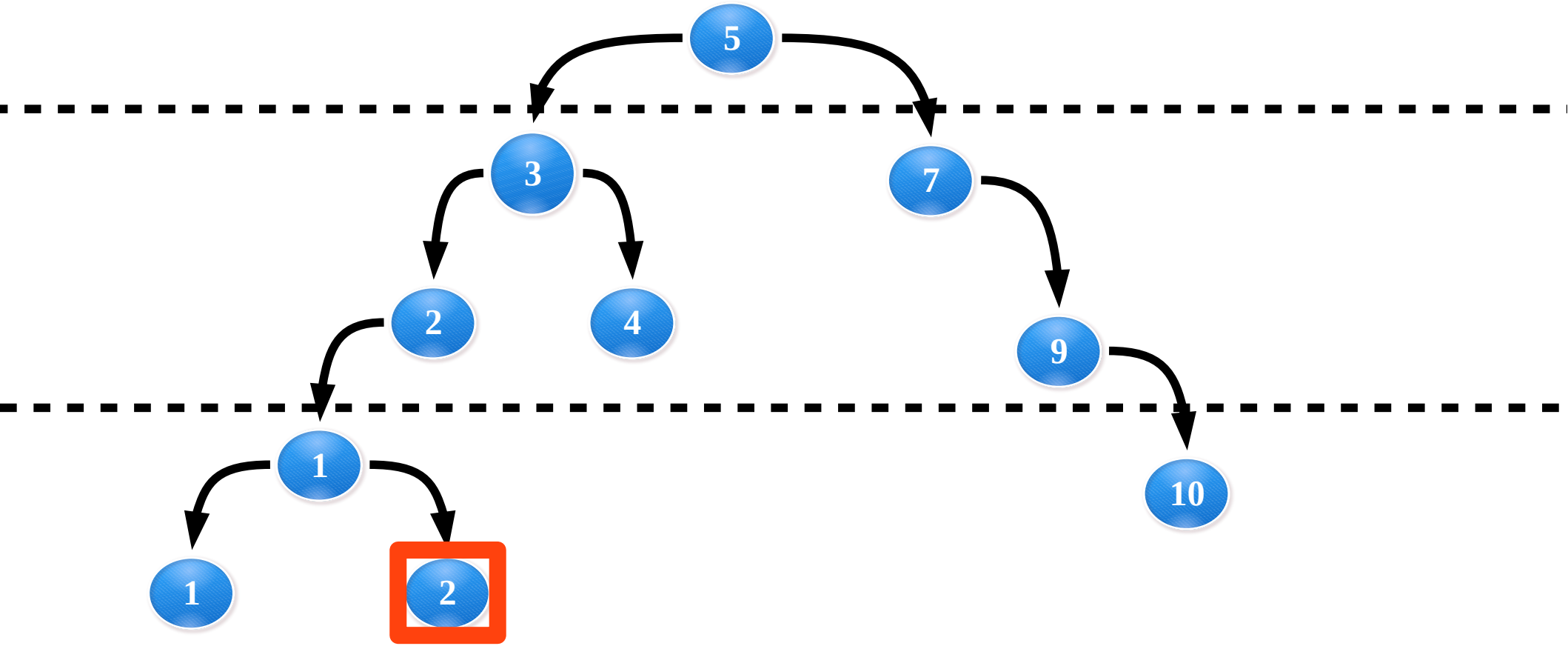
Statički alocirano stablo (Emde)

5	3	2	4	1	1		



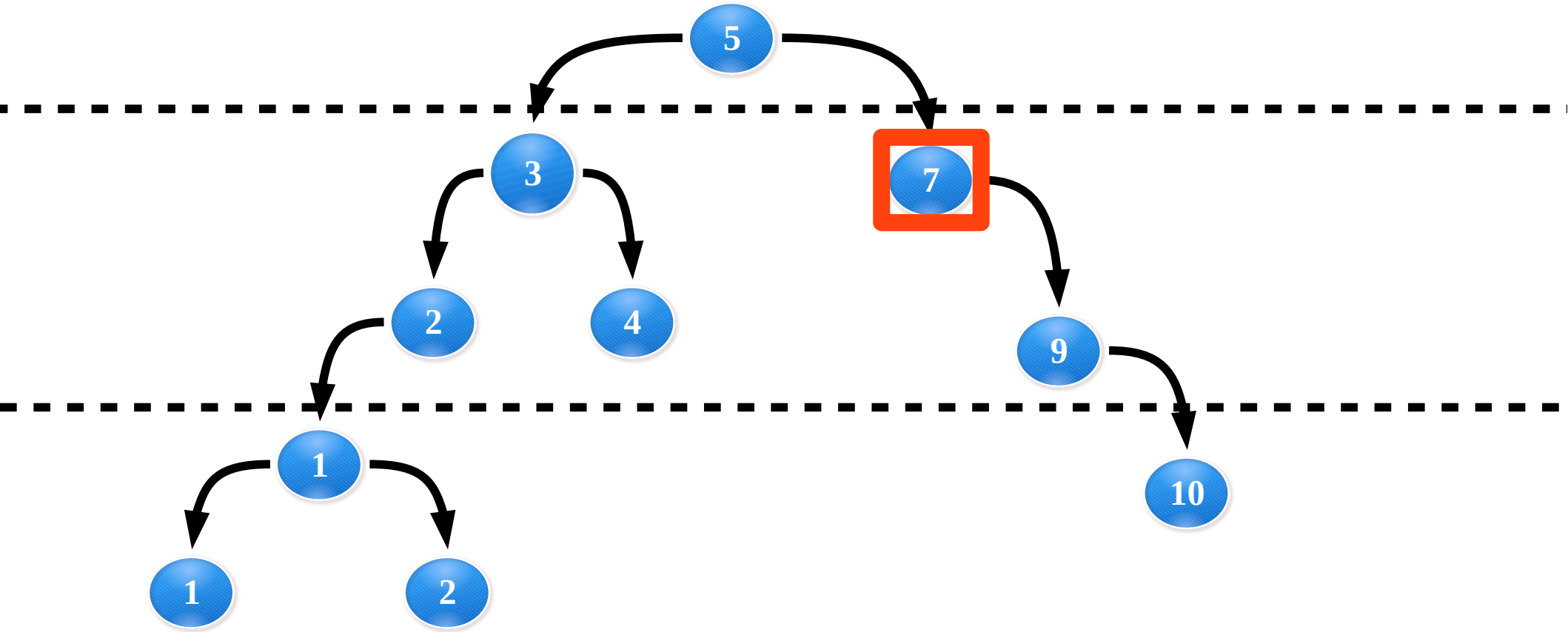
Statički alocirano stablo (Emde)

5	3	2	4	1	1	2		



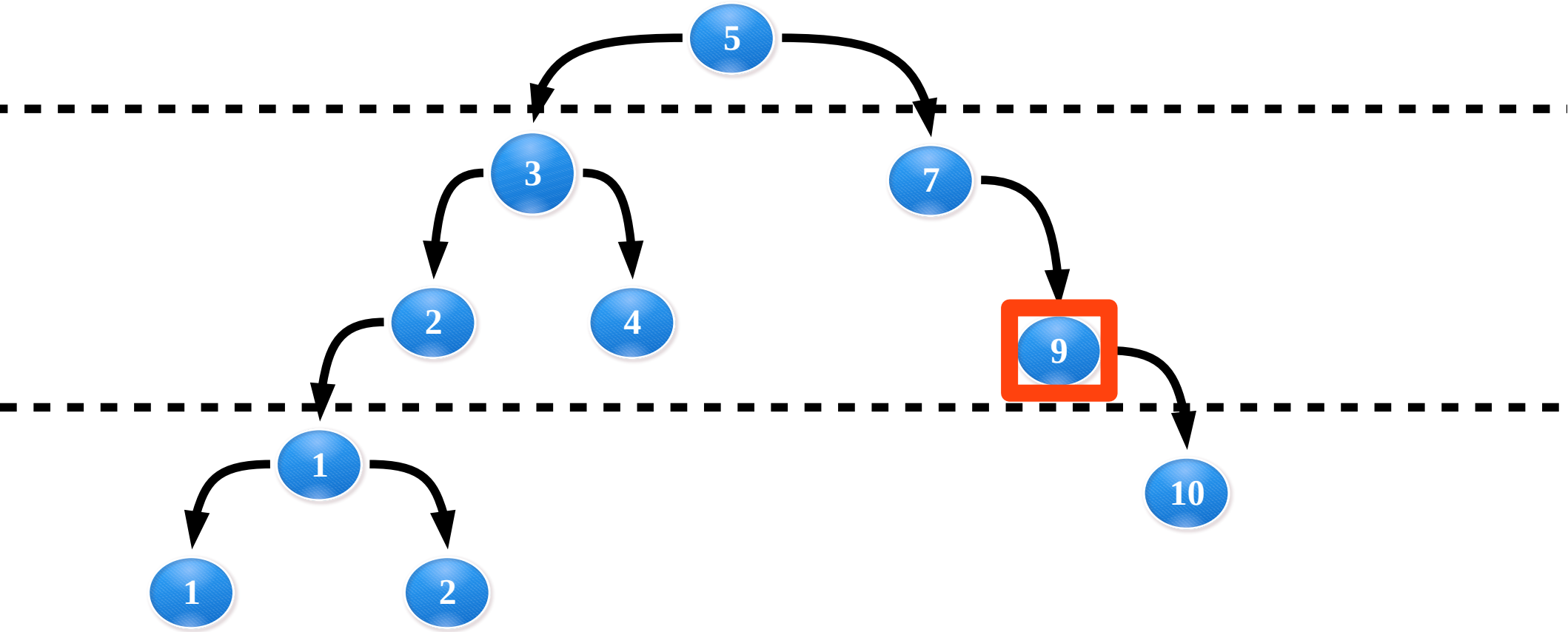
Statički alocirano stablo (Emde)

5	3	2	4	1	1	2	7		



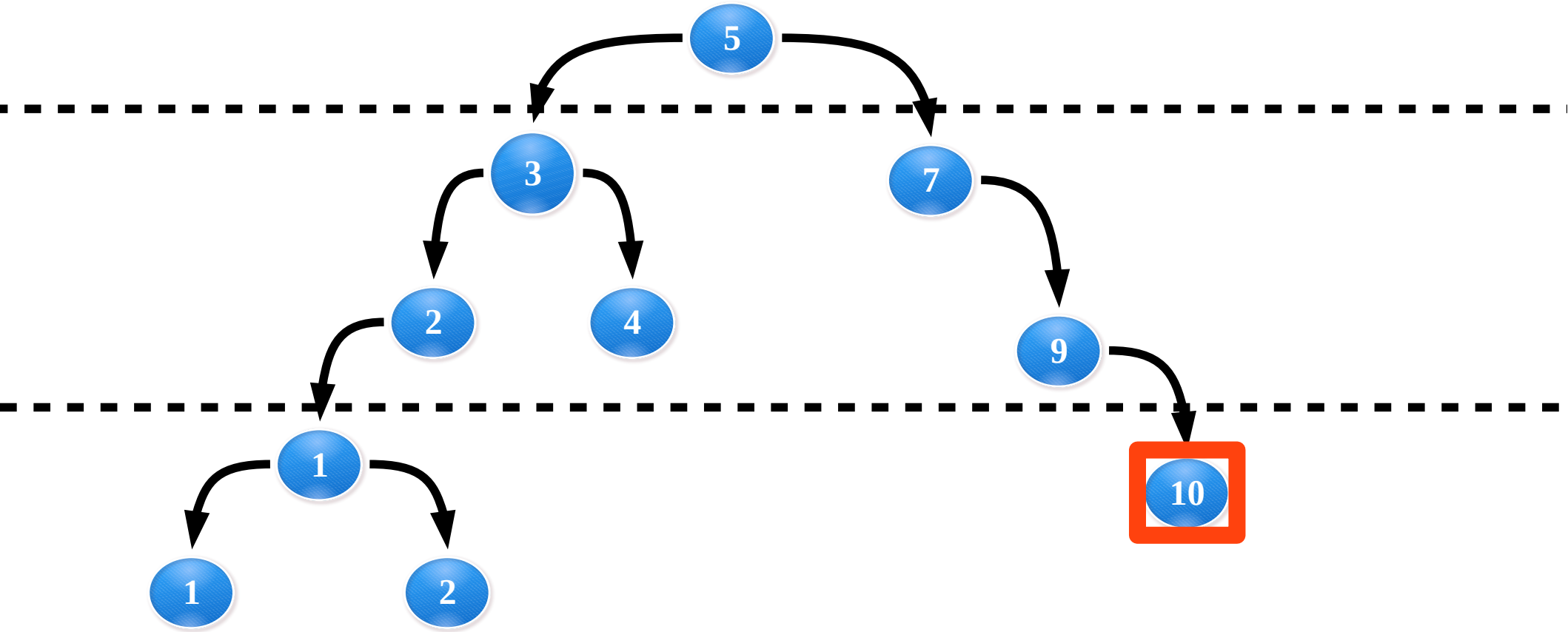
Statički alocirano stablo (Emde)

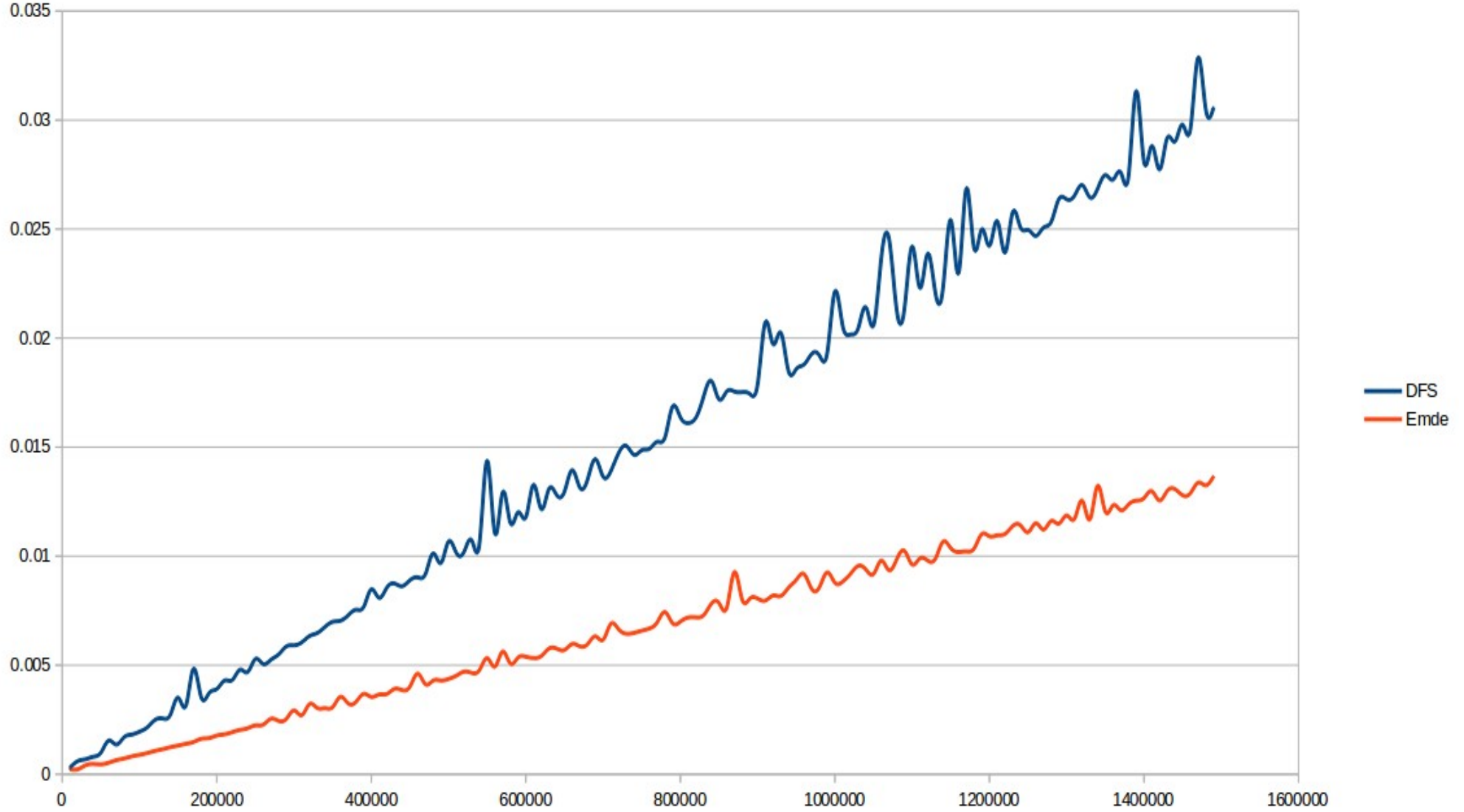
5	3	2	4	1	1	2	7	9	



Statički alocirano stablo (Emde)

5	3	2	4	1	1	2	7	9	10




0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
200000
400000
600000
800000
1000000
1200000
1400000
1600000
DFS
Emde

Linkovi

http://people.csail.mit.edu/bradley/papers/Kasheff04.pdf

http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-851-advanced-data-structures-spring-2012/lecture-videos/session-8-cache-oblivious-structures-i/

http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-851-advanced-data-structures-spring-2012/lecture-videos/session-9-cache-oblivious-structures-ii/