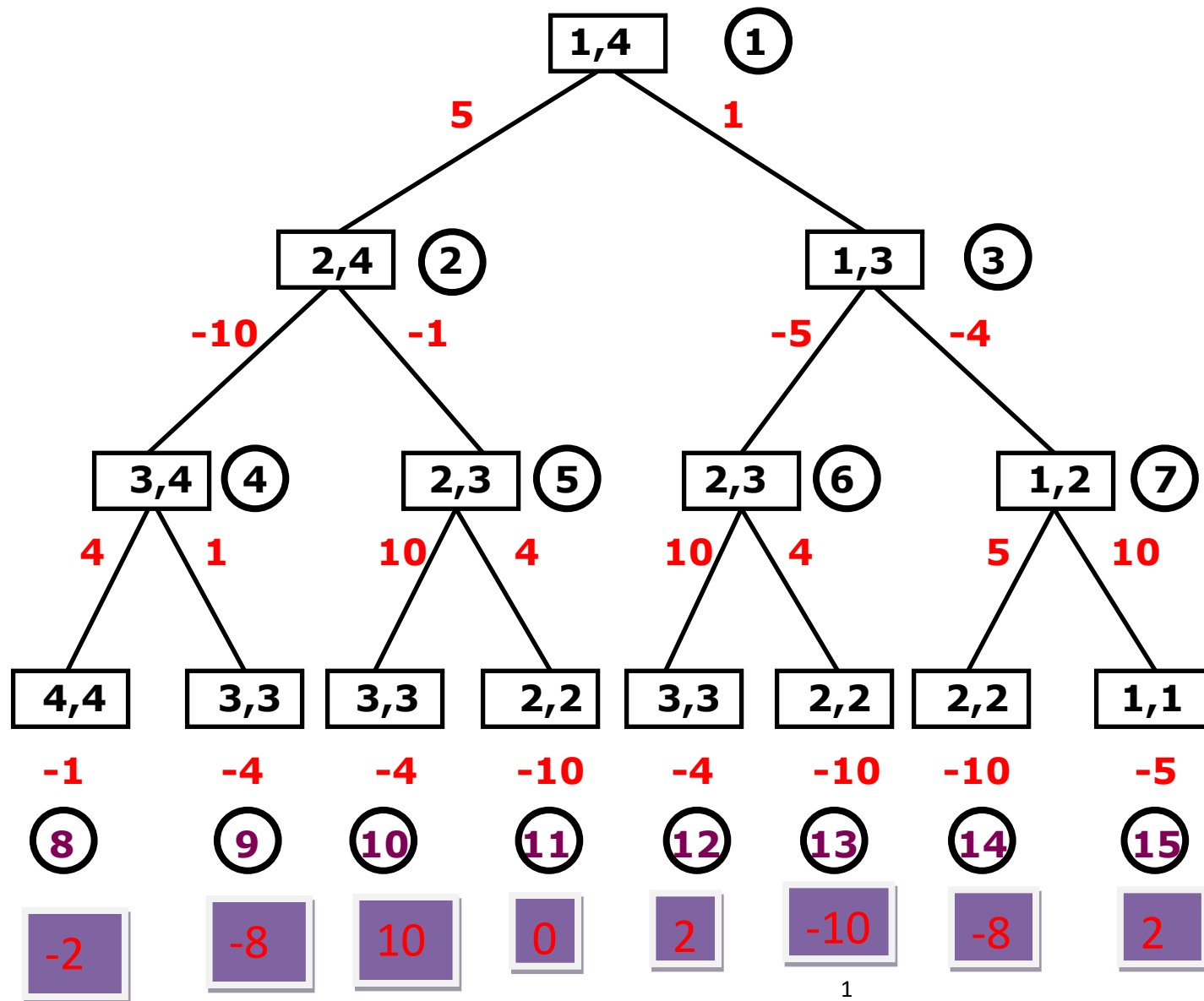
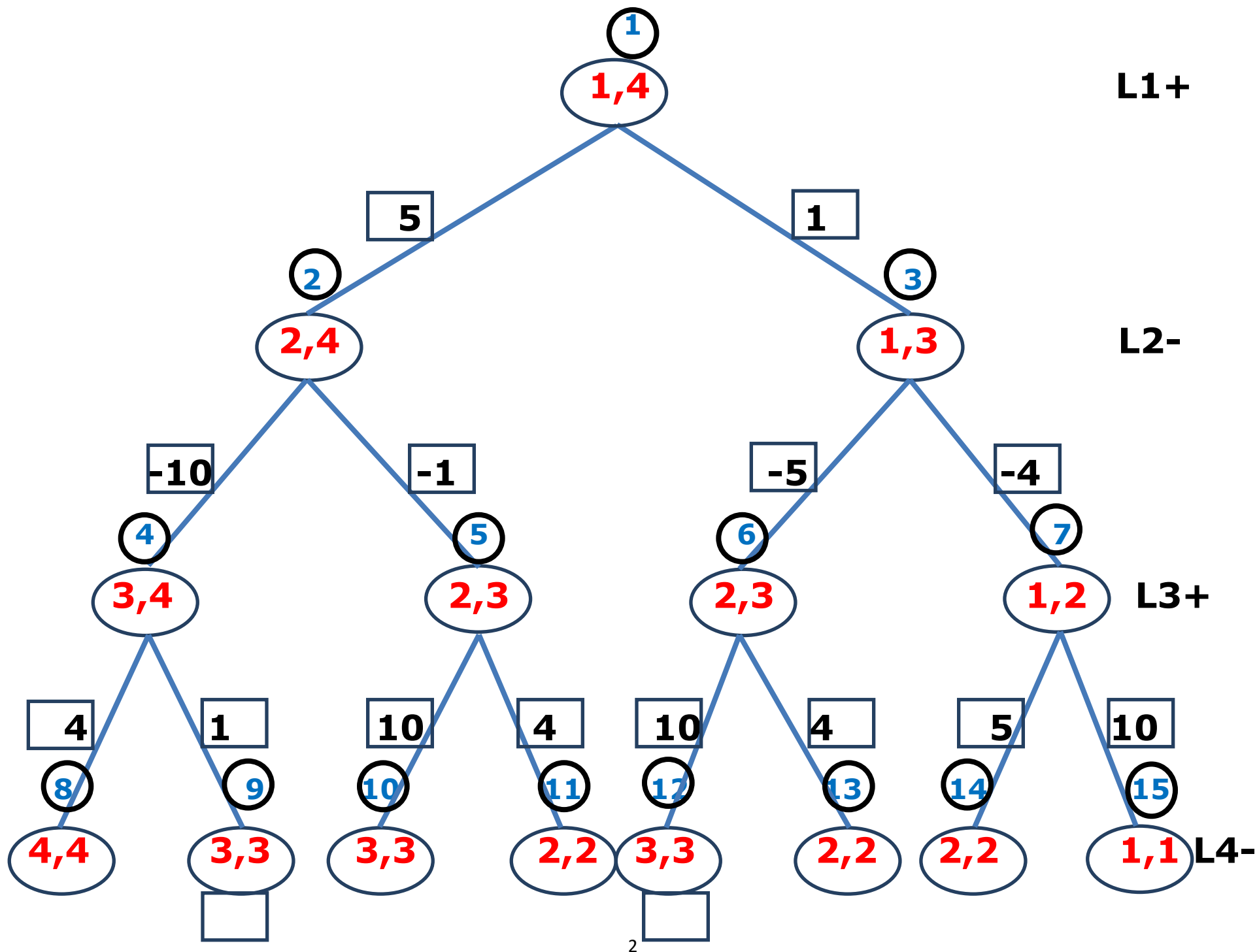


**int arr[] = {<sup>1 2 3 4</sup>5,10,4,1};**





<b>-1</b>	<b>-4</b>	<b>-4</b>	<b>-10</b>	<b>-4</b>	<b>-10</b>	<b>-10</b>	<b>-5</b>
1,4	1	i = 1					
2,4	2	i = 2					
1,3	3						
3,4	4						
2,3	5						
2,3	6						
1,2	7						
4,4	8						
3,3	9						
3,3	10						
2,2	11						
3,3	12						
2,2	13						
2,2	14						
1,1	15						

```
nodes[leftChild(i)] = new Node(nodes[i].index1+1,nodes[i].index2);  
nodes[rightChild(i)] = new Node(nodes[i].index1,nodes[i].index2-1);
```

```
private int leftChild(int i){return 2*i;}  
private int rightChild(int i){return 2*i+1;}
```

$$\mathbf{-1 + 4 - 10 + 5 = -2}$$

$$\mathbf{-4 + 1 - 10 + 5 = -8}$$

$$\mathbf{-4 + 10 - 1 + 5 = +10}$$

$$\mathbf{-10 + 4 - 1 + 5 = -2}$$

$$\mathbf{-4 + 10 - 5 + 1 = +2}$$

$$\mathbf{-10 + 4 - 5 + 1 = -10}$$

$$\mathbf{-10 + 5 - 4 + 1 = -8}$$

$$\mathbf{-5 + 10 - 4 + 1 = +2}$$

**=====**

**1,4 → 2,4 → 2,3 → 3,3      S1 - S2 = 10**

**5,10,4,1      S1 = 5 + 10      S2 = 1 + 4**

**Indexes   nodes[]**

**i = 1   leftChild(i) = 2   rightChild(i) = 3  
i = 2   leftChild(i) = 4   rightChild(i) = 5  
i = 3   leftChild(i) = 6   rightChild(i) = 7  
i = 4   leftChild(i) = 8   rightChild(i) = 9  
i = 5   leftChild(i) = 10   rightChild(i) = 11  
i = 6   leftChild(i) = 12   rightChild(i) = 13  
i = 7   leftChild(i) = 14   rightChild(i) = 15**

**1)   [1,4]   diff1 = 0   diff2 = 0   yes1 = false   yes2 = false  
2)   [2,4]   diff1 = 0   diff2 = 0   yes1 = false   yes2 = false  
3)   [1,3]   diff1 = 0   diff2 = 0   yes1 = false   yes2 = false  
4)   [3,4]   diff1 = 0   diff2 = 0   yes1 = false   yes2 = false  
5)   [2,3]   diff1 = 0   diff2 = 0   yes1 = false   yes2 = false  
6)   [2,3]   diff1 = 0   diff2 = 0   yes1 = false   yes2 = false**

7)	[1,2]	diff1 = 0	diff2 = 0	yes1 = false	yes2 = false
8)	[4,4]	diff1 = 0	diff2 = 0	yes1 = false	yes2 = false
9)	[3,3]	diff1 = 0	diff2 = 0	yes1 = false	yes2 = false
10)	[3,3]	diff1 = 0	diff2 = 0	yes1 = false	yes2 = false
11)	[2,2]	diff1 = 0	diff2 = 0	yes1 = false	yes2 = false
12)	[3,3]	diff1 = 0	diff2 = 0	yes1 = false	yes2 = false
13)	[2,2]	diff1 = 0	diff2 = 0	yes1 = false	yes2 = false
14)	[2,2]	diff1 = 0	diff2 = 0	yes1 = false	yes2 = false
15)	[1,1]	diff1 = 0	diff2 = 0	yes1 = false	yes2 = false

```

// constructor
public GameTree(int []arr){
    prices = new int[arr.length+1];
    for (int i=1; i<prices.length; i++){
        prices[i] = arr[i-1];
    }
    // create tree
    int len = (int)Math.pow(2, arr.length);
    int n = arr.length;
    nodes = new Node[len];
    nodes[1] = new Node(1,n);

    for (int i=1; i<nodes.length/2; i++){
        System.out.println("i = " + i + " leftChild(i) = " + leftChild(i) + " rightChild(i) = "
            + rightChild(i));
        nodes[leftChild(i)] = new Node(nodes[i].index1+1,nodes[i].index2);
        nodes[rightChild(i)] = new Node(nodes[i].index1,nodes[i].index2-1);
    }
}

```



```

public int getWinPrice(){
    // set difference for leaves
    int level = nodes.length/2;
    for(int i=nodes.length/2; i<nodes.length; i++){
        nodes[i].diff1 = -prices[nodes[i].index1];
        nodes[i].diff2 = -prices[nodes[i].index2];
    }
    printTree();
    //***** 1 *****
    //*****
    // set difference to parents
    level = level/2;
    int factor = 1;
    int winPrice = 0, diff1=0, diff2=0;
    int m1 = 0, m2 = 0;

    for(int i=nodes.length-1; i>1; i=i-2){
        int par = parent(i);
        diff1 = nodes[i].diff1 + factor*prices[nodes[par].index2];
        diff2 = nodes[i].diff2 + factor*prices[nodes[par].index2];
    }
}

```

```

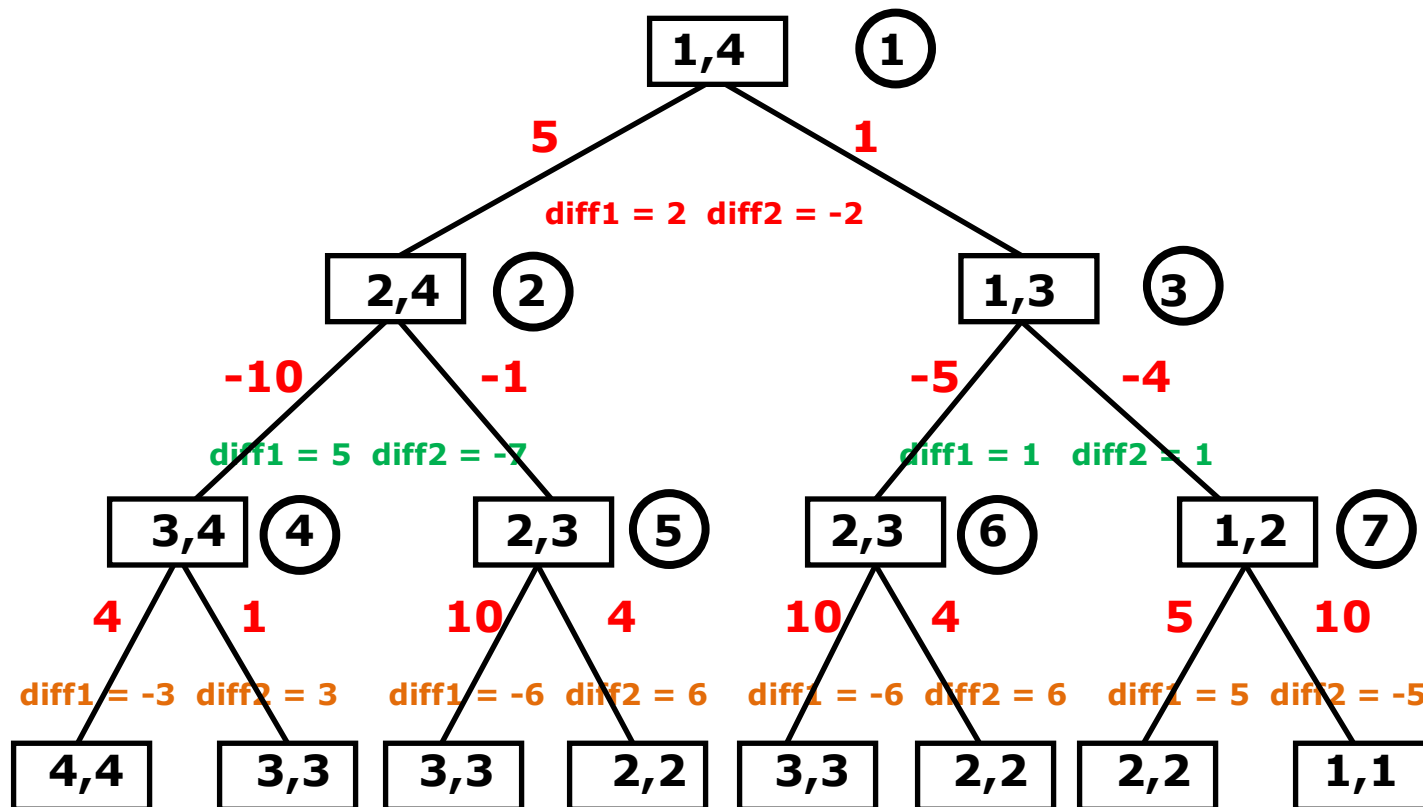
    m1 = Math.max(diff1, diff2);
    if (factor>0) m1 = Math.min(diff1, diff2);
    nodes[par].diff1 = m1;
    diff1 = nodes[i-1].diff1 + factor*prices[nodes[par].index1];
    diff2 = nodes[i-1].diff2 + factor*prices[nodes[par].index1];
    m2 = Math.max(diff1, diff2);
    if (factor>0) m2 = Math.min(diff1, diff2);
    nodes[par].diff2 = m2;
    // build path
    if (factor>0){
        if (m1 >= m2) nodes[par].yes1 = true;
        else nodes[par].yes2 = true;
    }
    else{
        if (m1 >= m2) nodes[par].yes2 = true;
        else nodes[par].yes1 = true;
    }
    if (par == level){
        level = level/2;
        factor = -factor;
        //System.out.println(i+", factor: "+factor+", level: "+level);
    }
}

//printTree();
winPrice = Math.max(m1, m2);
return winPrice;

```

}

**int arr[]** = {<sup>1</sup>5,<sup>2</sup>10,<sup>3</sup>4,<sup>4</sup>1};



level 1 +

level 2 -

level 3 +

level 4 -

-1	-4	-4	-10	-4	-10	-10	-5
8	9	10	11	12	13	14	15

- 1) [1,4] diff1 = 0 diff2 = 0 yes1 = false yes2 = false
- 2) [2,4] diff1 = 0 diff2 = 0 yes1 = false yes2 = false
- 3) [1,3] diff1 = 0 diff2 = 0 yes1 = false yes2 = false
- 4) [3,4] diff1 = 0 diff2 = 0 yes1 = false yes2 = false
- 5) [2,3] diff1 = 0 diff2 = 0 yes1 = false yes2 = false
- 6) [2,3] diff1 = 0 diff2 = 0 yes1 = false yes2 = false
- 7) [1,2] diff1 = 0 diff2 = 0 yes1 = false yes2 = false
- 8) [4,4] diff1 = -1 diff2 = -1 yes1 = false yes2 = false**
- 9) [3,3] diff1 = -4 diff2 = -4 yes1 = false yes2 = false**
- 10) [3,3] diff1 = -4 diff2 = -4 yes1 = false yes2 = false**
- 11) [2,2] diff1 = -10 diff2 = -10 yes1 = false yes2 = false**
- 12) [3,3] diff1 = -4 diff2 = -4 yes1 = false yes2 = false**
- 13) [2,2] diff1 = -10 diff2 = -10 yes1 = false yes2 = false**
- 14) [2,2] diff1 = -10 diff2 = -10 yes1 = false yes2 = false**
- 15) [1,1] diff1 = -5 diff2 = -5 yes1 = false yes2 = false**

factor: 1, level: 4  
**i = 15**  
par = 7  
diff1 = 5   diff2 = 5  
m1 = 5  
m1 = 5  
diff1 = -5   diff2 = -5  
m2 = -5  
m2 = -5  
par = 7   level = 4  
**i = 13**  
par = 6  
diff1 = -6   diff2 = -6  
m1 = -6  
m1 = -6  
diff1 = 6   diff2 = 6  
m2 = 6  
m2 = 6  
par = 6   level = 4  
**i = 11**  
par = 5  
diff1 = -6   diff2 = -6  
m1 = -6

m1 = -6  
diff1 = 6 diff2 = 6  
m2 = 6  
m2 = 6  
par = 5 level = 4  
**i = 9**  
par = 4  
diff1 = -3 diff2 = -3  
m1 = -3  
m1 = -3  
diff1 = 3 diff2 = 3  
m2 = 3  
m2 = 3  
par = 4 level = 4  
i = 9, factor: -1, level: 2

- 1) [1,4] diff1 = 0 diff2 = 0 yes1 = false yes2 = false
- 2) [2,4] diff1 = 0 diff2 = 0 yes1 = false yes2 = false
- 3) [1,3] diff1 = 0 diff2 = 0 yes1 = false yes2 = false
- 4) [3,4] diff1 = -3 diff2 = 3 yes1 = false yes2 = true**
- 5) [2,3] diff1 = -6 diff2 = 6 yes1 = false yes2 = true**
- 6) [2,3] diff1 = -6 diff2 = 6 yes1 = false yes2 = true**
- 7) [1,2] diff1 = 5 diff2 = -5 yes1 = true yes2 = false**
- 8) [4,4] diff1 = -1 diff2 = -1 yes1 = false yes2 = false**
- 9) [3,3] diff1 = -4 diff2 = -4 yes1 = false yes2 = false**
- 10) [3,3] diff1 = -4 diff2 = -4 yes1 = false yes2 = false**
- 11) [2,2] diff1 = -10 diff2 = -10 yes1 = false yes2 = false**

12) [3,3] diff1 = -4 diff2 = -4 yes1 = false yes2 = false  
13) [2,2] diff1 = -10 diff2 = -10 yes1 = false yes2 = false  
14) [2,2] diff1 = -10 diff2 = -10 yes1 = false yes2 = false  
15) [1,1] diff1 = -5 diff2 = -5 yes1 = false yes2 = false

=====

**i = 7**

par = 3

diff1 = 1 diff2 = -9

m1 = 1

m1 = 1

diff1 = -11 diff2 = 1

m2 = 1

m2 = 1

par = 3 level = 2

**i = 5**

par = 2

diff1 = -7 diff2 = 5

m1 = 5

m1 = 5

diff1 = -13 diff2 = -7

m2 = -7

m2 = -7

par = 2 level = 2

i = 5, factor: 1, level: 1

1) [1,4] diff1 = 0 diff2 = 0 yes1 = false yes2 = false

**2) [2,4] diff1 = 5 diff2 = -7 yes1 = false yes2 = true**

3) [1,3] diff1 = 1 diff2 = 1 yes1 = false yes2 = true  
 4) [3,4] diff1 = -3 diff2 = 3 yes1 = false yes2 = true  
 5) [2,3] diff1 = -6 diff2 = 6 yes1 = false yes2 = true  
 6) [2,3] diff1 = -6 diff2 = 6 yes1 = false yes2 = true  
 7) [1,2] diff1 = 5 diff2 = -5 yes1 = true yes2 = false  
 8) [4,4] diff1 = -1 diff2 = -1 yes1 = false yes2 = false  
 9) [3,3] diff1 = -4 diff2 = -4 yes1 = false yes2 = false  
 10) [3,3] diff1 = -4 diff2 = -4 yes1 = false yes2 = false  
 11) [2,2] diff1 = -10 diff2 = -10 yes1 = false yes2 = false  
 12) [3,3] diff1 = -4 diff2 = -4 yes1 = false yes2 = false  
 13) [2,2] diff1 = -10 diff2 = -10 yes1 = false yes2 = false  
 14) [2,2] diff1 = -10 diff2 = -10 yes1 = false yes2 = false  
 15) [1,1] diff1 = -5 diff2 = -5 yes1 = false yes2 = false  
 =====

**i = 3**  
 par = 1  
 diff1 = 2 diff2 = 2  
 m1 = 2  
 m1 = 2  
 diff1 = 10 diff2 = -2  
 m2 = 10  
 m2 = -2  
 par = 1 level = 1  
 i = 3, factor: -1, level: 0

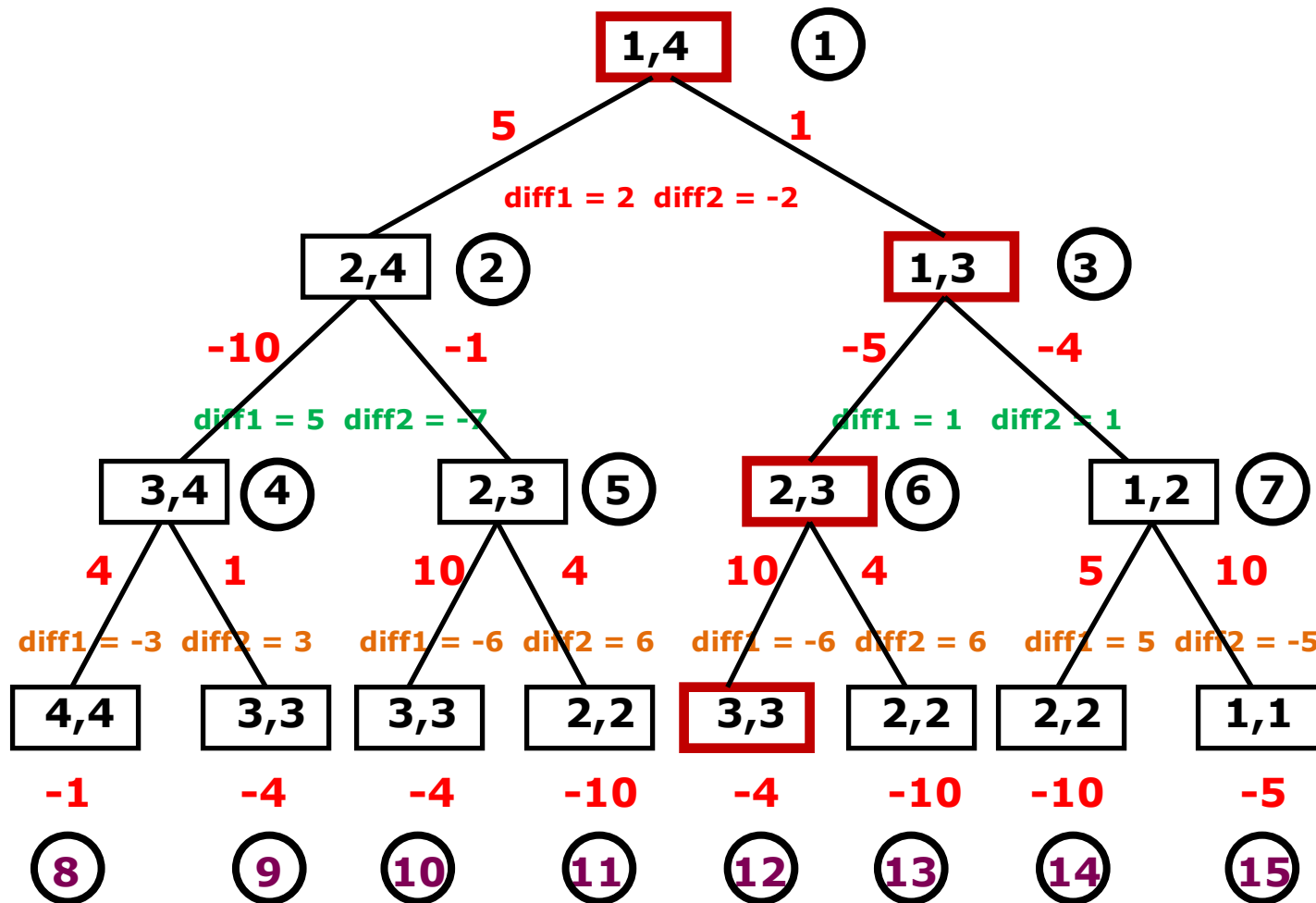
1) [1,4] diff1 = 2 diff2 = -2 yes1 = true yes2 = false  
 2) [2,4] diff1 = 5 diff2 = -7 yes1 = false yes2 = true



- 3) [1,3] diff1 = 1 diff2 = 1 yes1 = false yes2 = true
  - 4) [3,4] diff1 = -3 diff2 = 3 yes1 = false yes2 = true
  - 5) [2,3] diff1 = -6 diff2 = 6 yes1 = false yes2 = true
  - 6) [2,3] diff1 = -6 diff2 = 6 yes1 = false yes2 = true
  - 7) [1,2] diff1 = 5 diff2 = -5 yes1 = true yes2 = false
  - 8) [4,4] diff1 = -1 diff2 = -1 yes1 = false yes2 = false
  - 9) [3,3] diff1 = -4 diff2 = -4 yes1 = false yes2 = false
  - 10) [3,3] diff1 = -4 diff2 = -4 yes1 = false yes2 = false
  - 11) [2,2] diff1 = -10 diff2 = -10 yes1 = false yes2 = false
  - 12) [3,3] diff1 = -4 diff2 = -4 yes1 = false yes2 = false
  - 13) [2,2] diff1 = -10 diff2 = -10 yes1 = false yes2 = false
  - 14) [2,2] diff1 = -10 diff2 = -10 yes1 = false yes2 = false
  - 15) [1,1] diff1 = -5 diff2 = -5 yes1 = false yes2 = false
- =====

1 2 3 4

**int arr[] = {5,10,4,1};**



printPath()

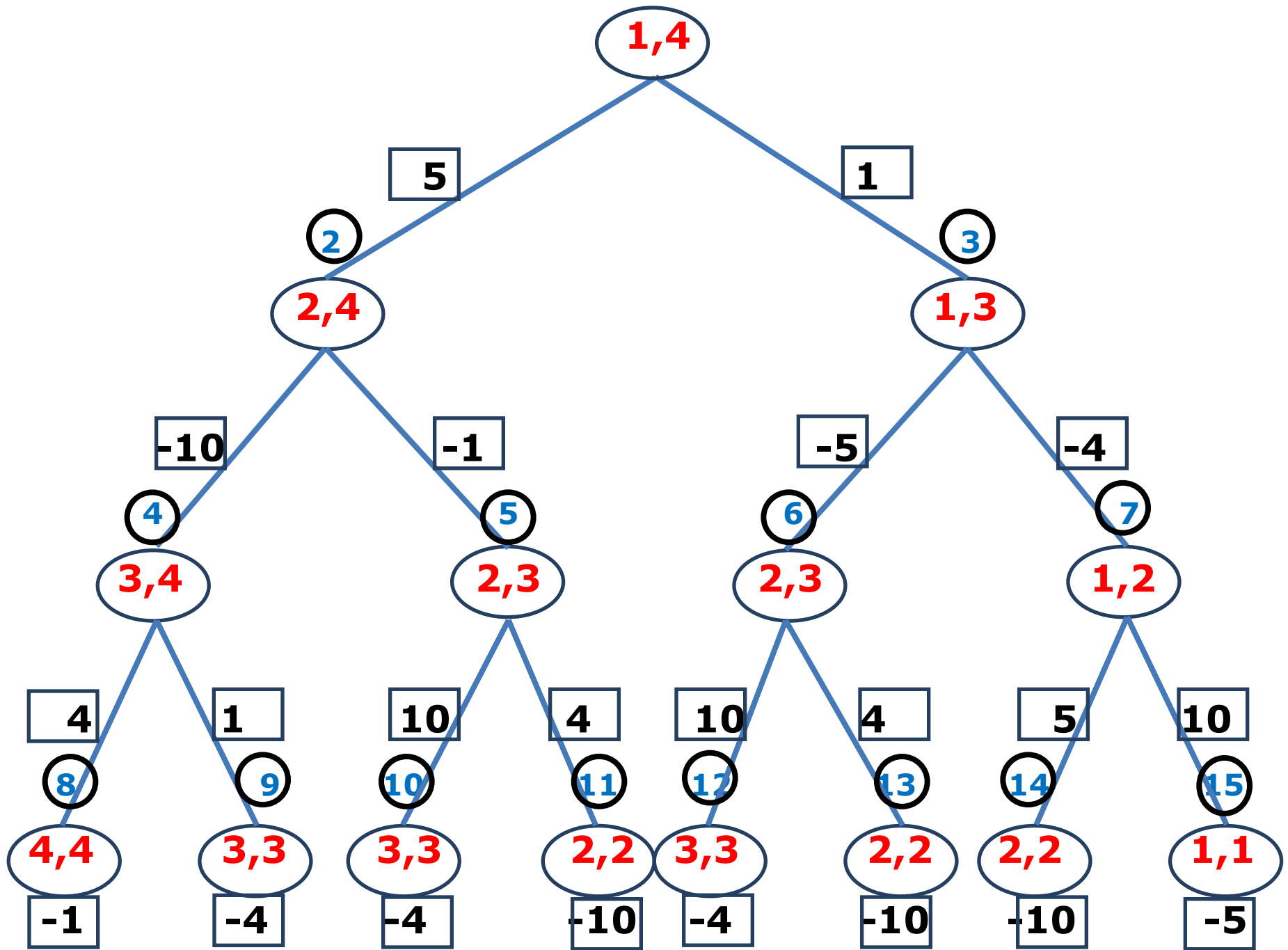
```
child = 1    nodes[child].yes1 = true  
child = 3    nodes[child].yes2 = true  
child = 6    nodes[child].yes2 = true  
path: 4->1->2->3  
winPrice: 2
```

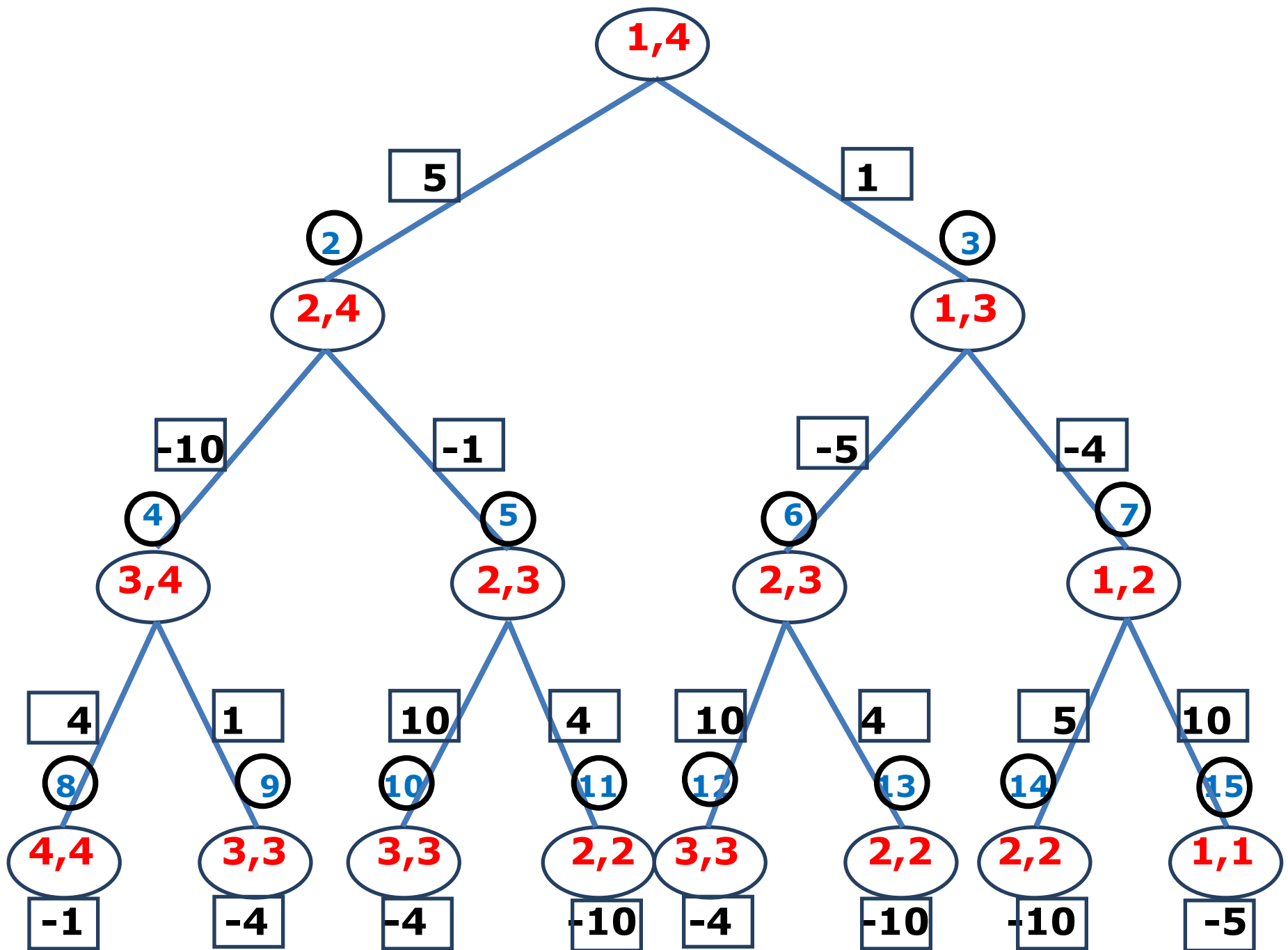
```
public void printPath(){  
    String s = "";  
    int child = 1;
```

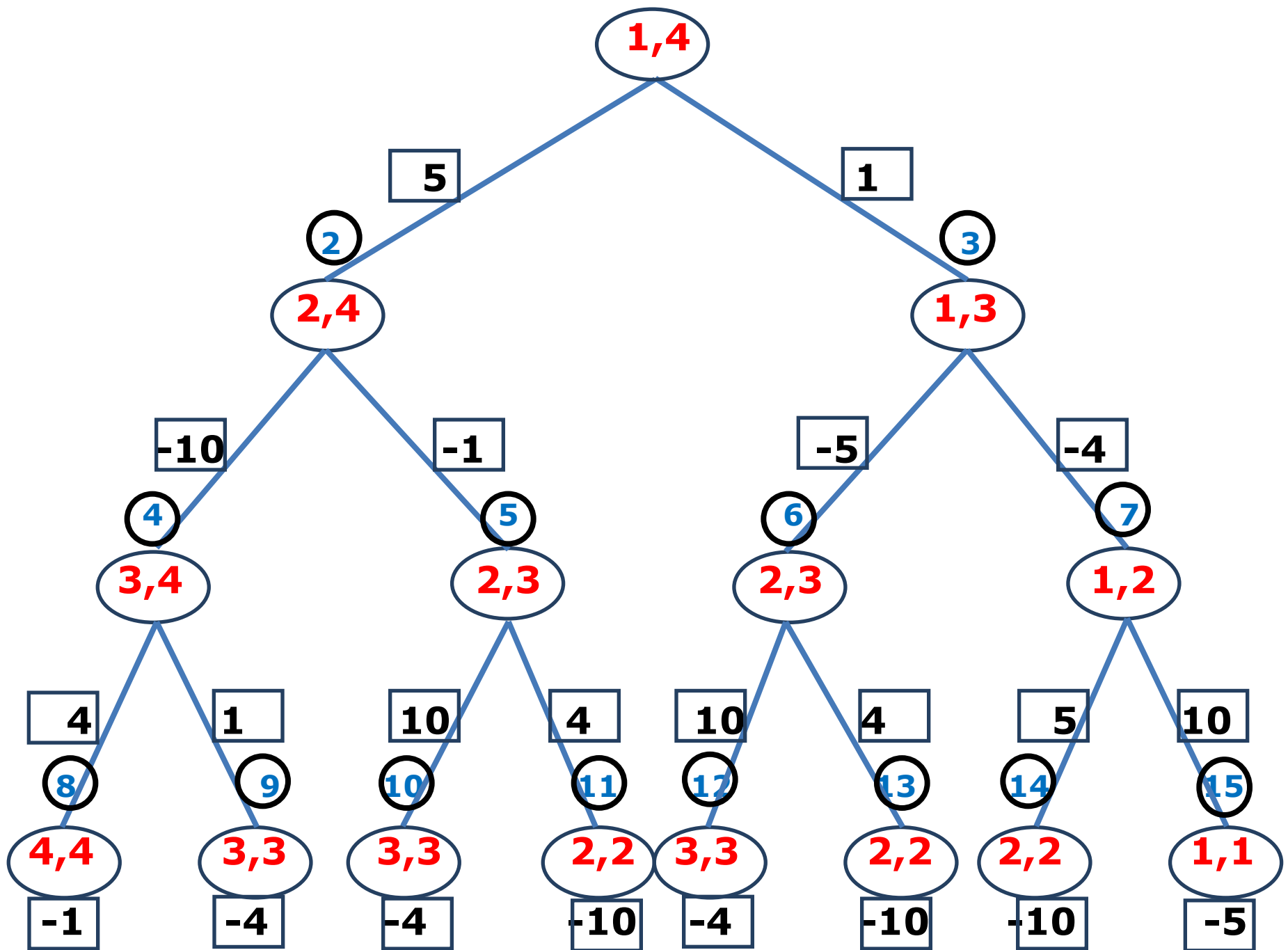
```

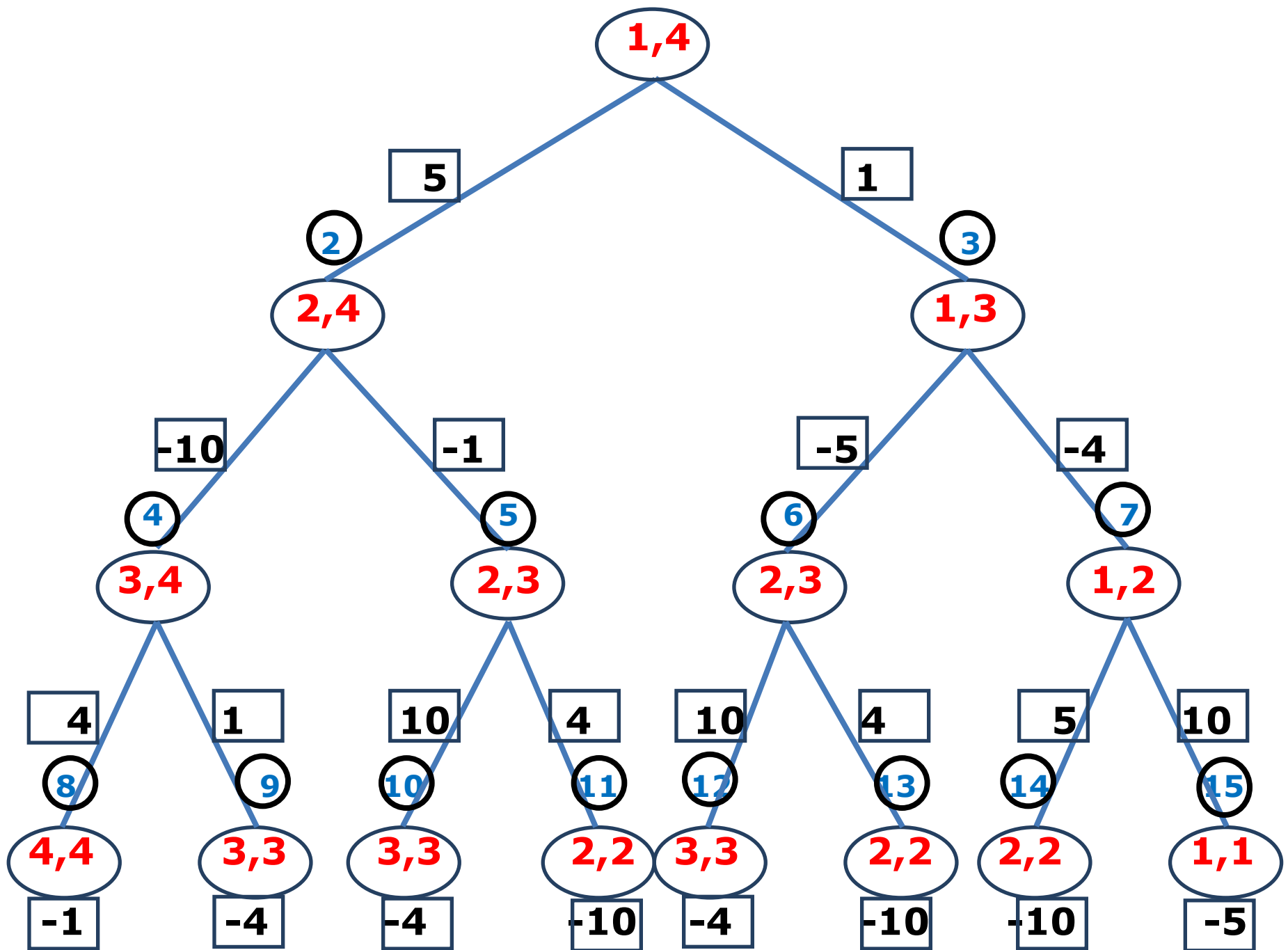
for (int i=2; i<prices.length; i++){
    if (nodes[child].yes1){
        System.out.println("child = " + child + "    nodes[child].yes1 = " +
                           nodes[child].yes1);
        s = s + nodes[child].index2 + "->";
        child = rightChild(child);
    }
    else if (nodes[child].yes2){
        System.out.println("child = " + child + "    nodes[child].yes2 = " +
                           nodes[child].yes2);
        s = s + nodes[child].index1 + "->";
        child = leftChild(child);
    }
}
s = s + nodes[child].index2;
System.out.println("path: " + s);
}

```

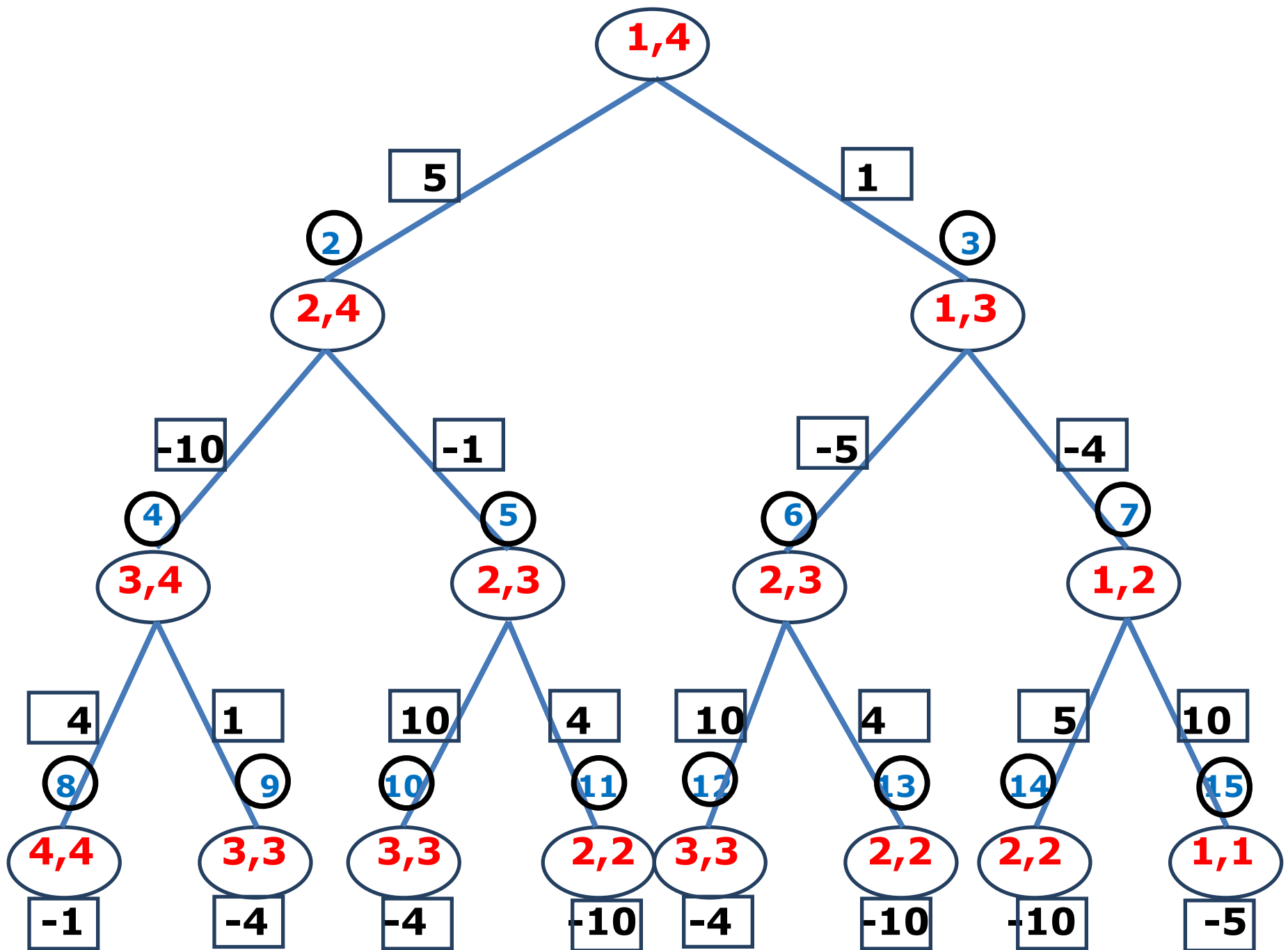




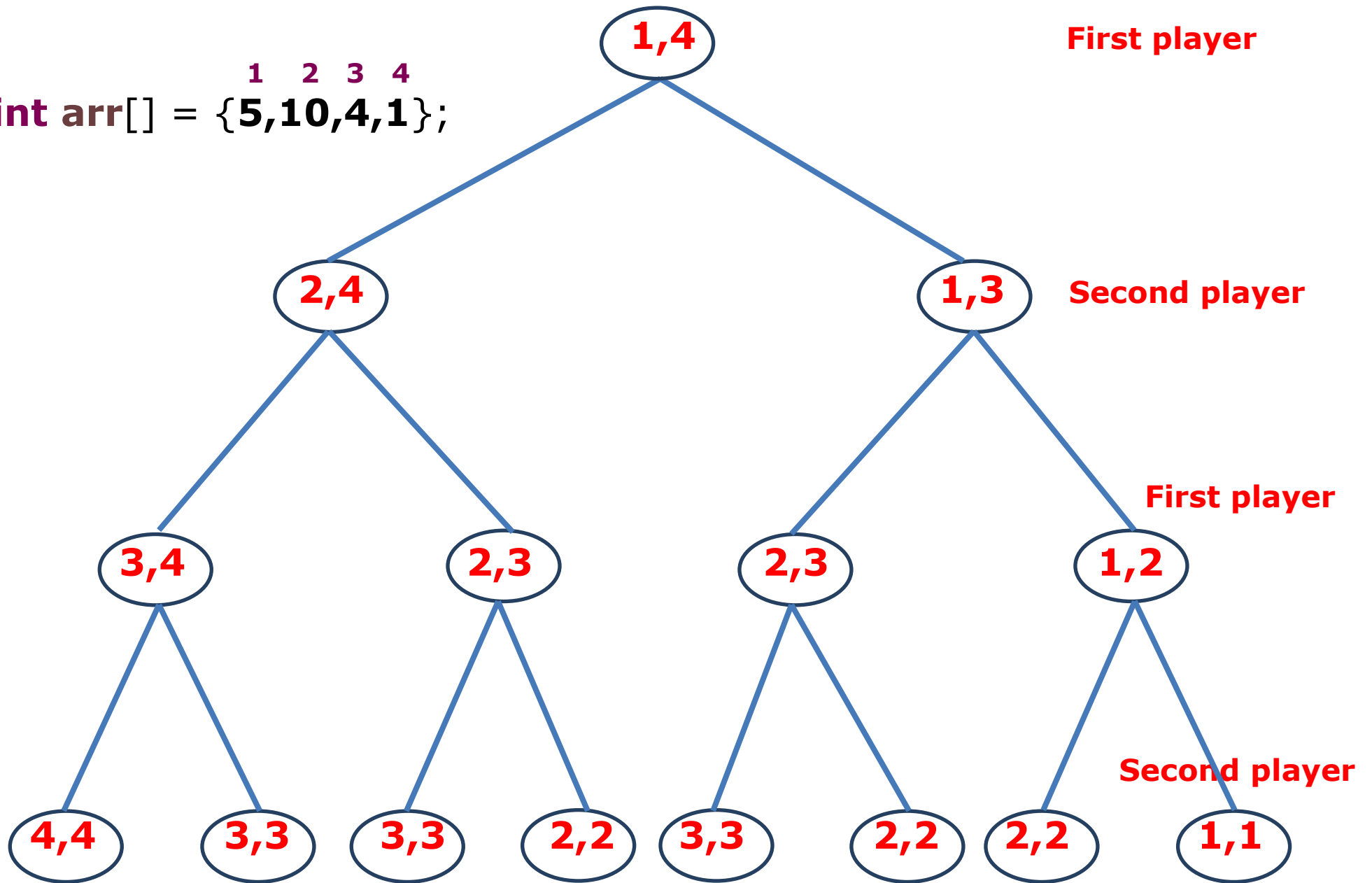








`int arr[] = {15,210,34,41};`

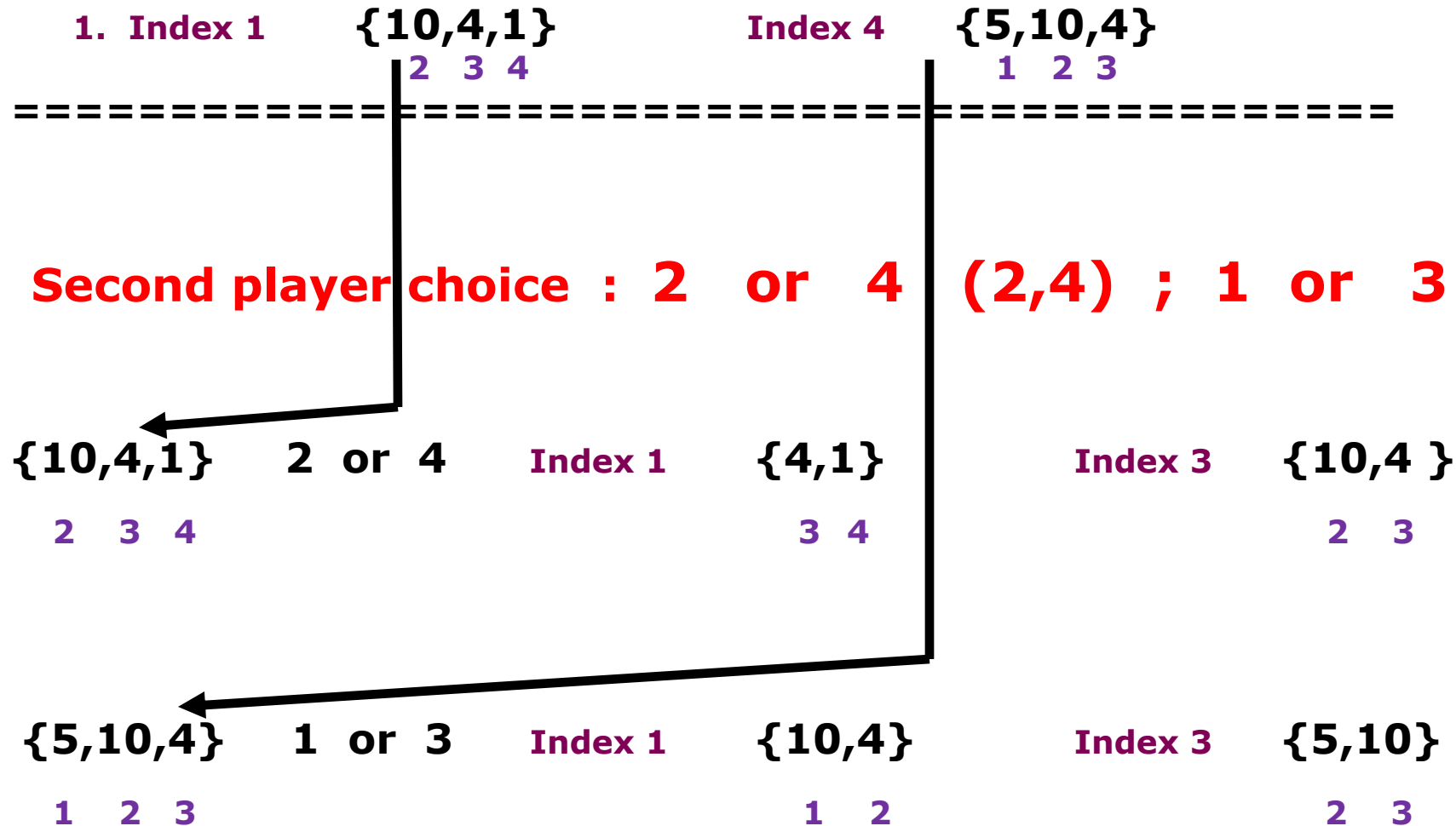


index            1   2   3   4

**int arr[] = {5,10,4,1};**

=====

**First player choice : 1 or 4        (1,4)**



=====

**First player choice : 3 or 4; 2 or 3;; 2 or 3; 1 or 2**