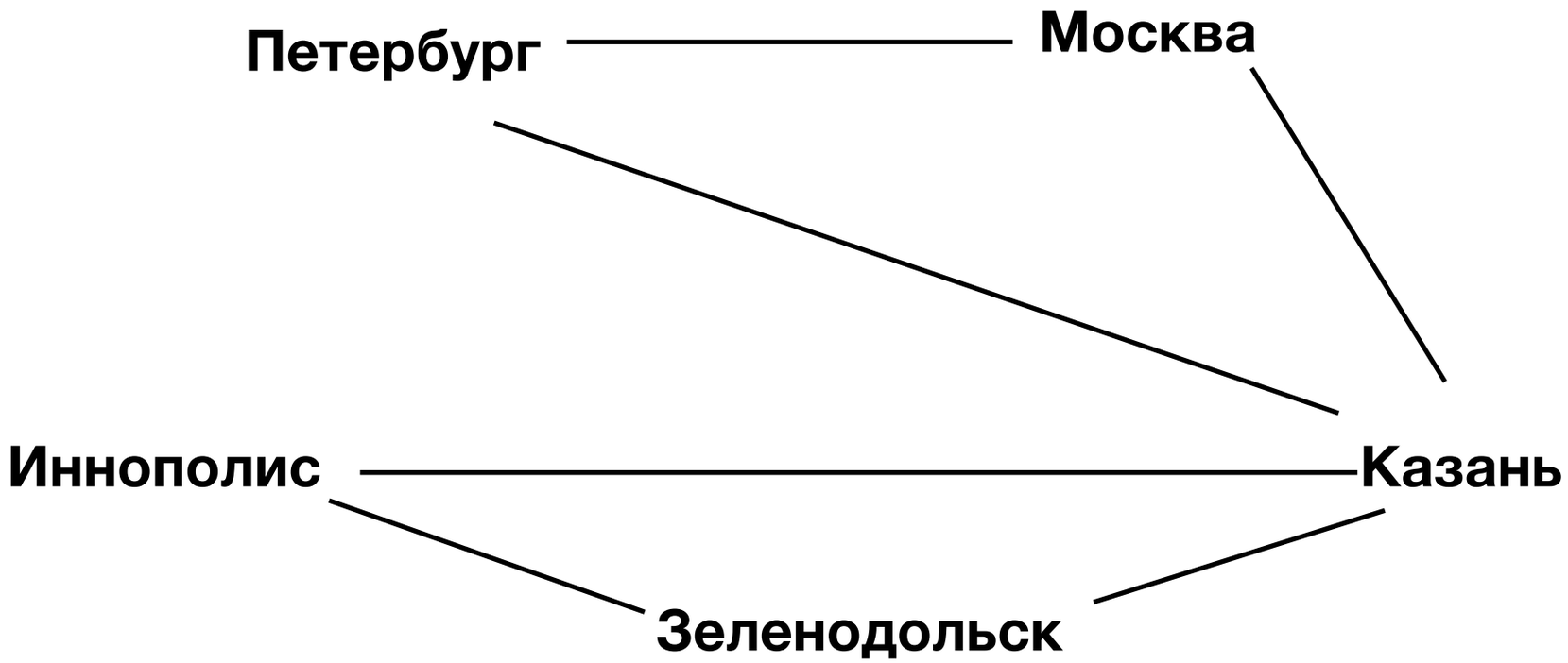
Графы

Disclamer

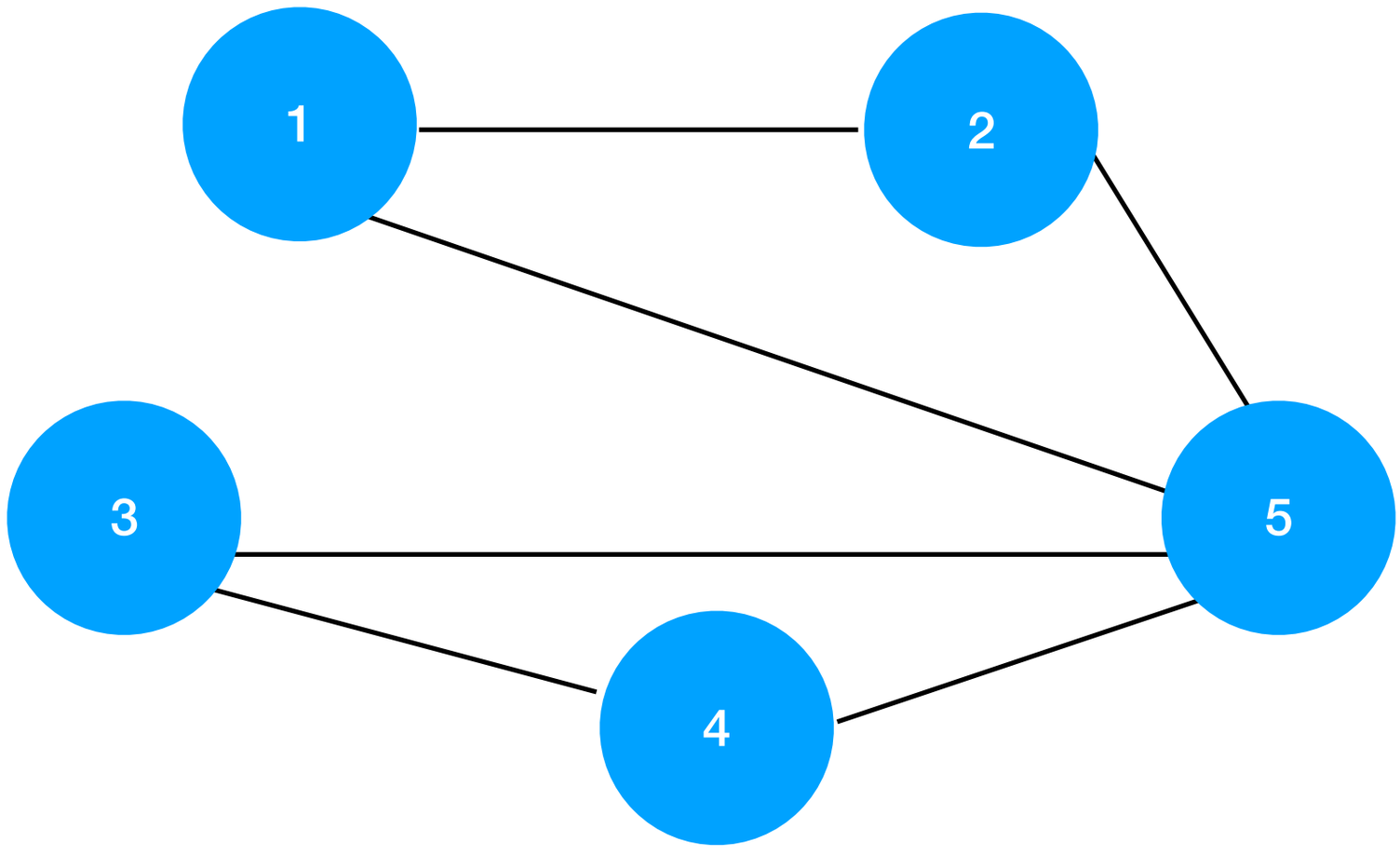
1) **Если вы ученик, то, возможно, многое будет непонятно. Но все хорошо: осознание придет со временем. Почитайте еще и попрактикуйтесь. Теория без практики - абсолютно бесполезна.**

2) **В материалах могут содержаться неточности, так как они предназначены для объяснения концептов ученикам практически “на пальцах”.**

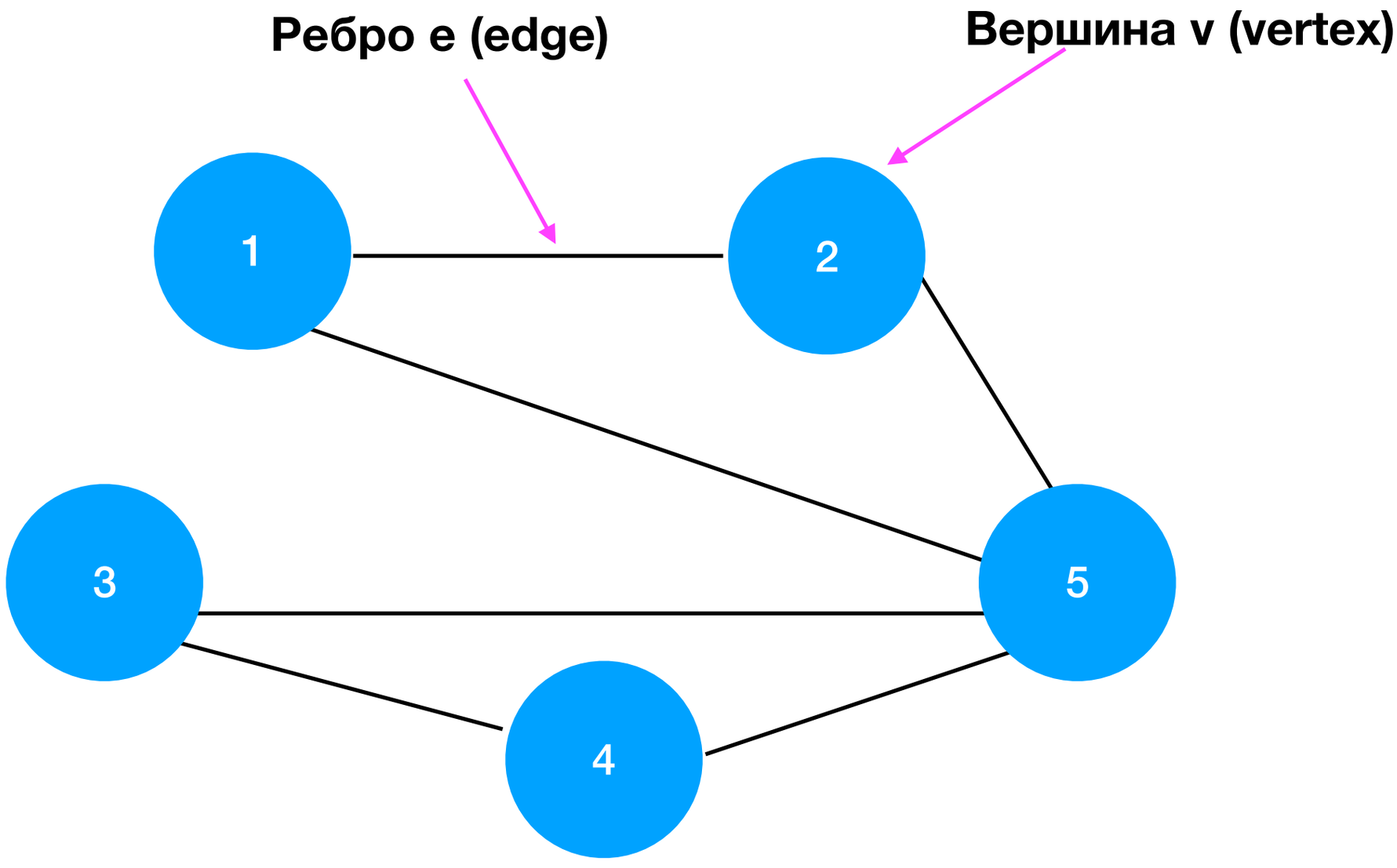
Пример



Теория

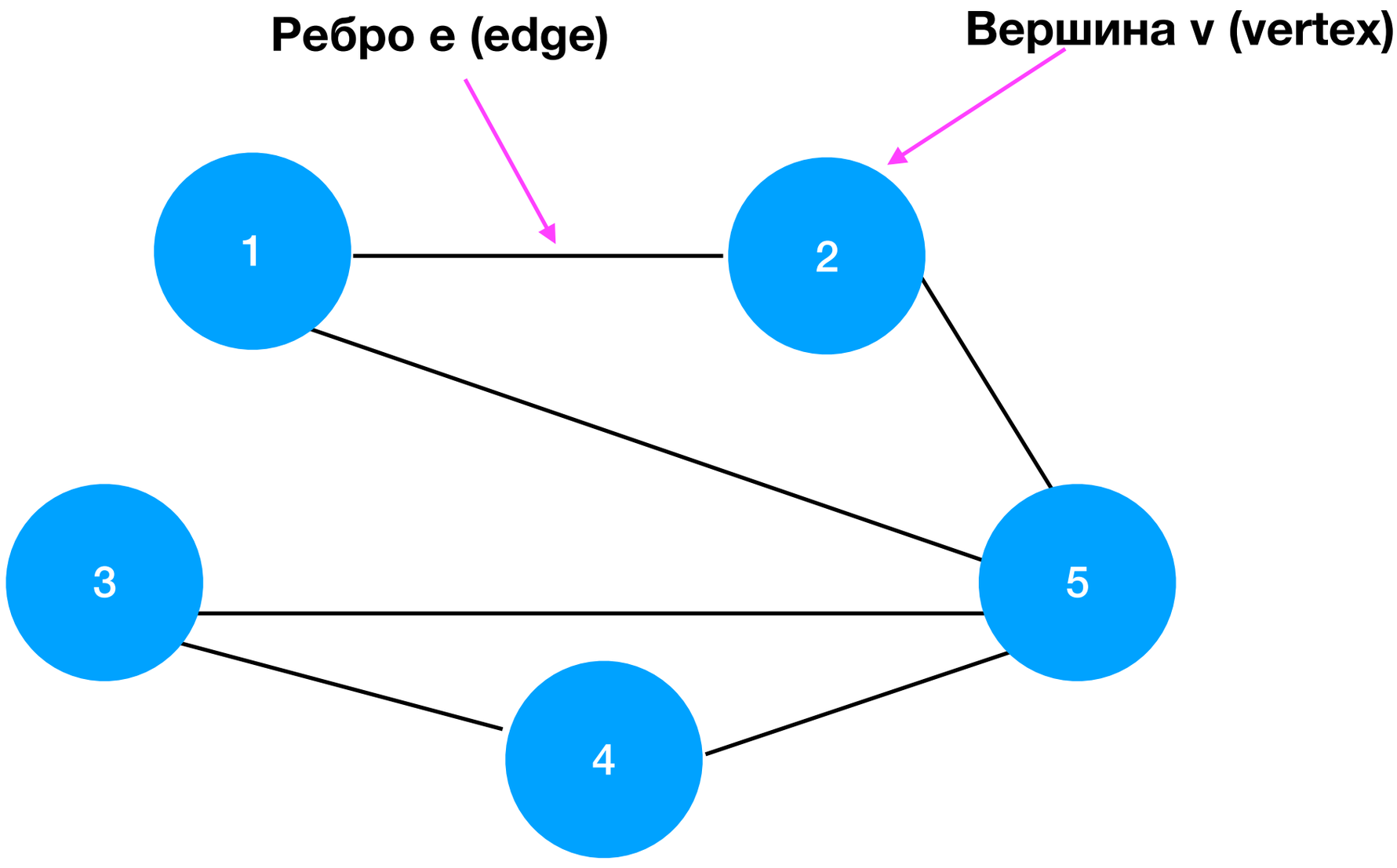


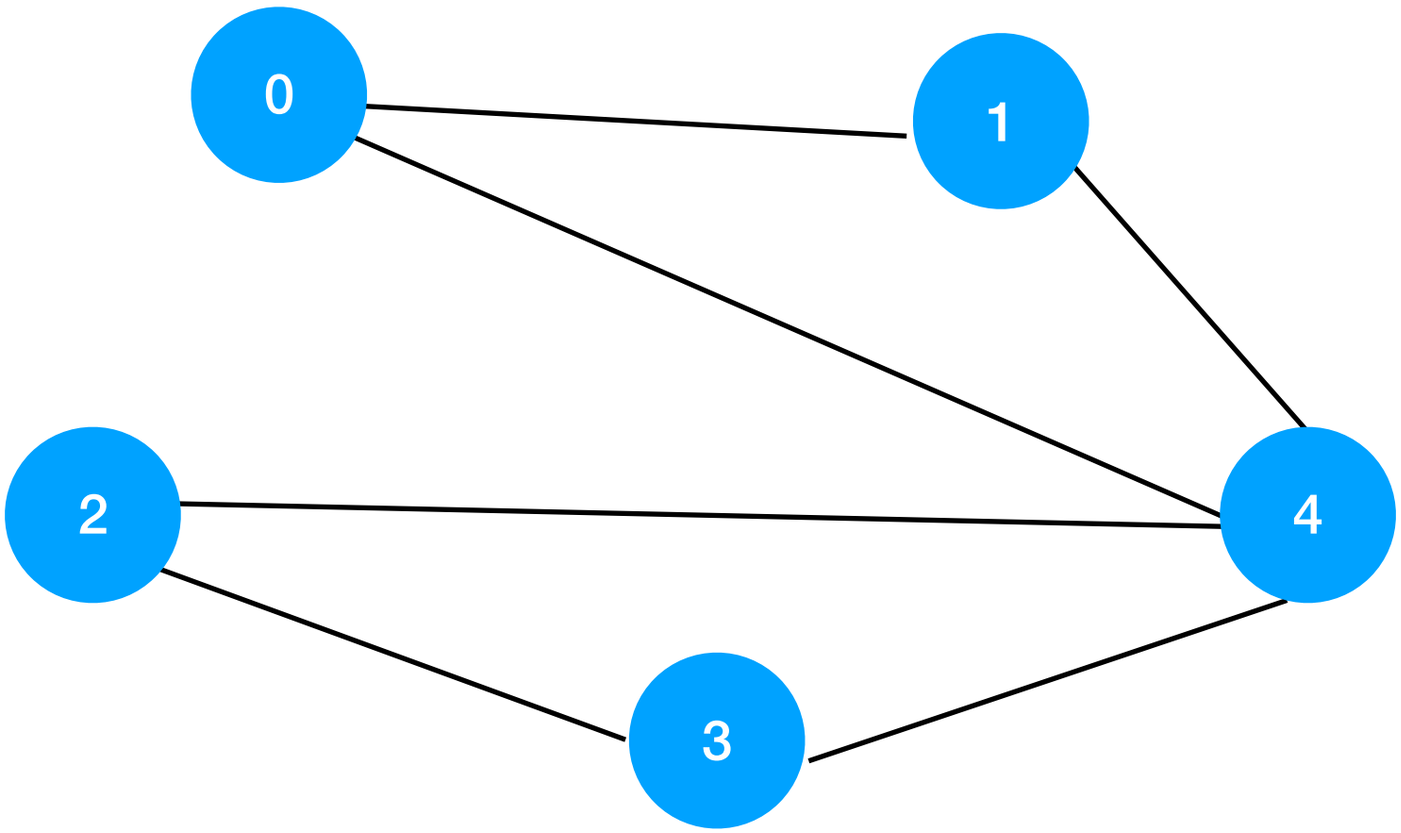
Неориентированный граф

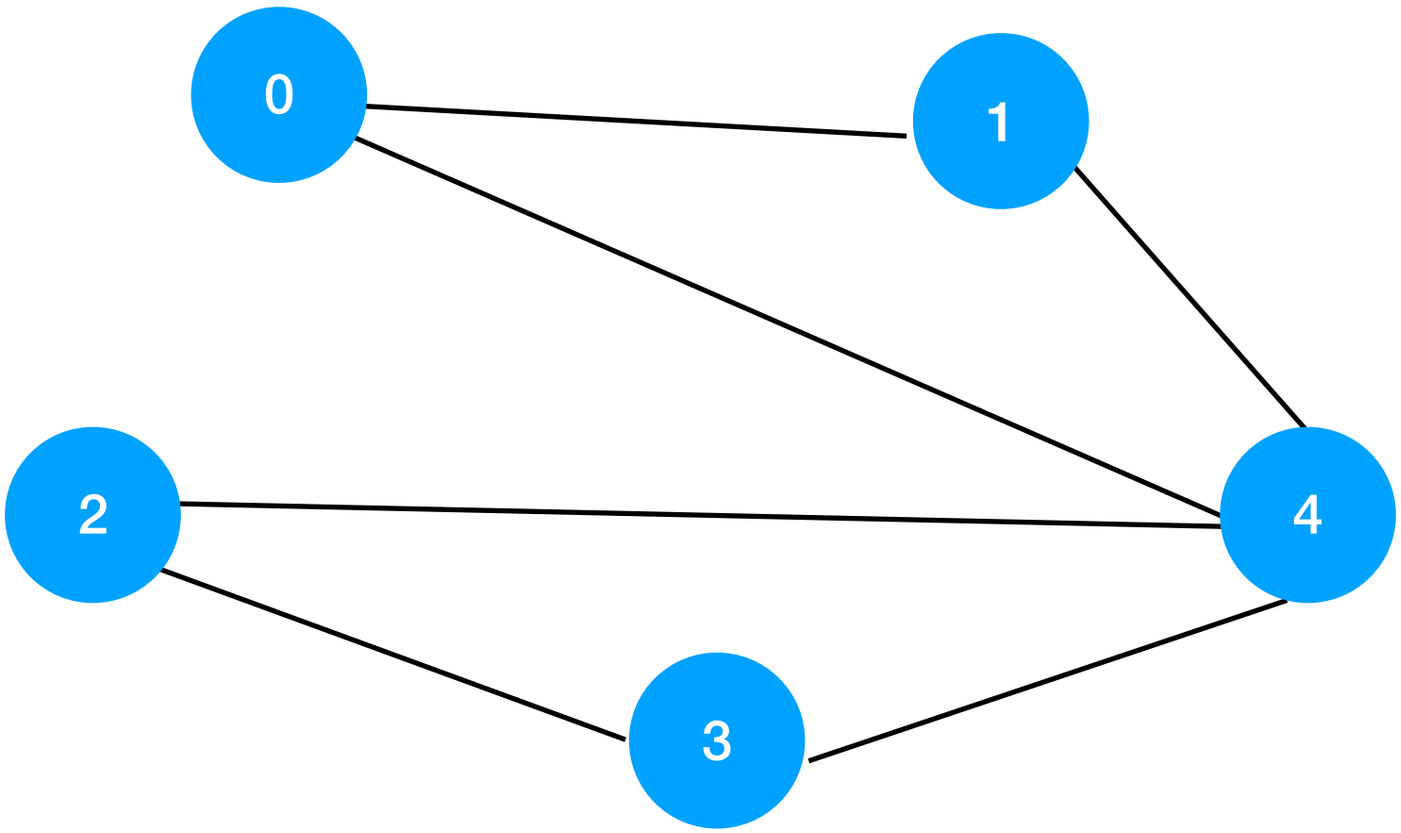


V - множество вершин

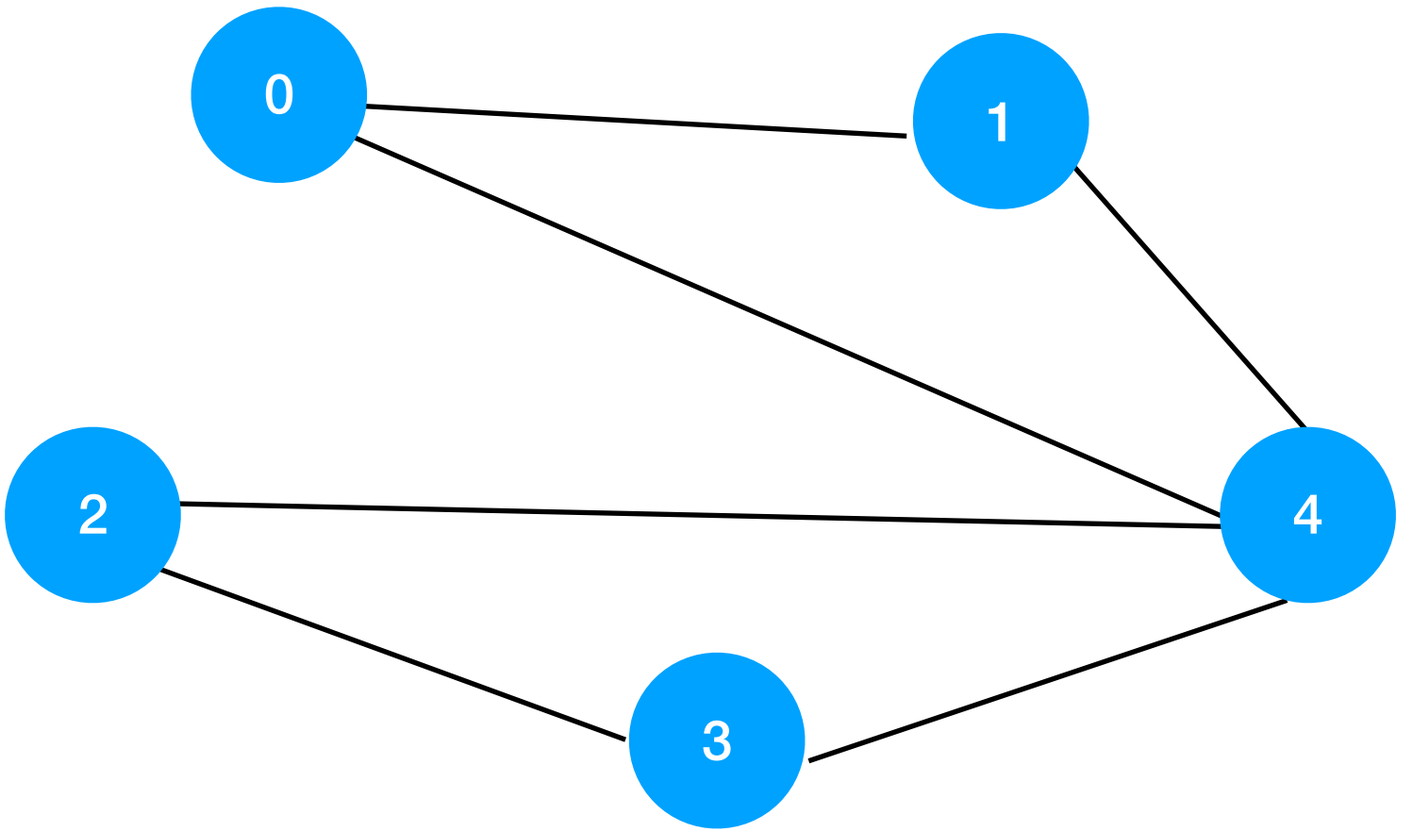
E - множество ребер



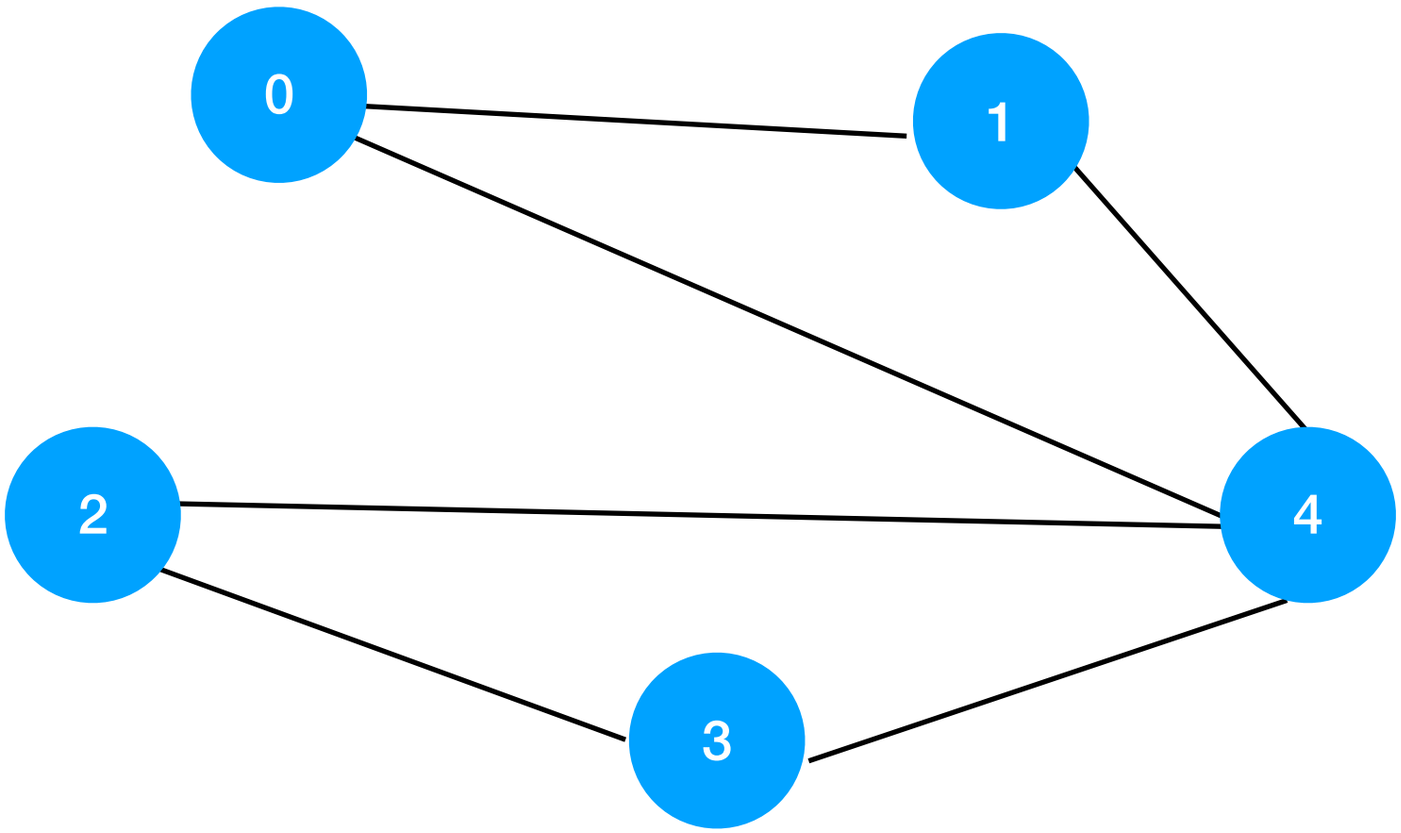

0
1
2
4
3



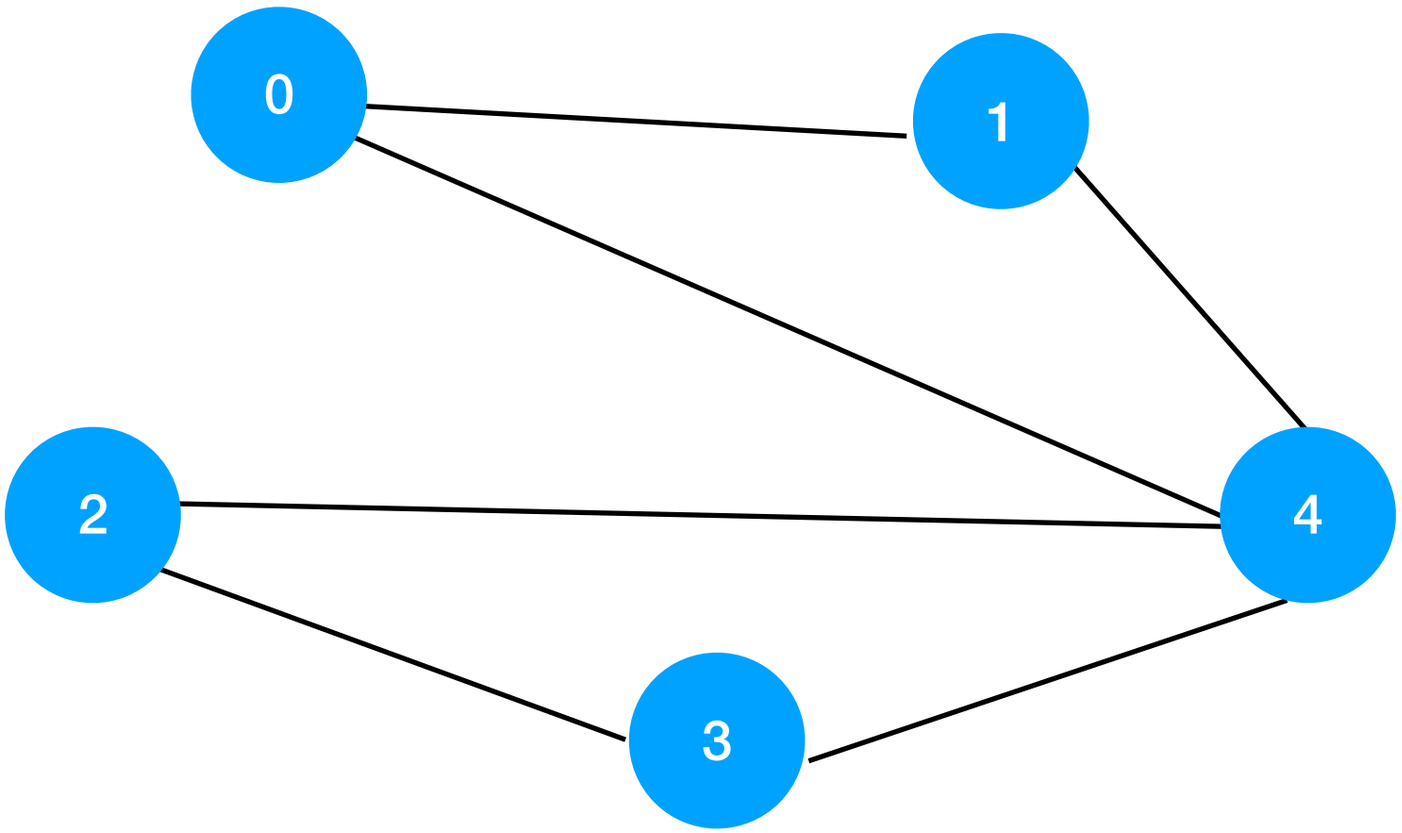
	0	1	2	3	4
0					
1					
2					
3					
4					



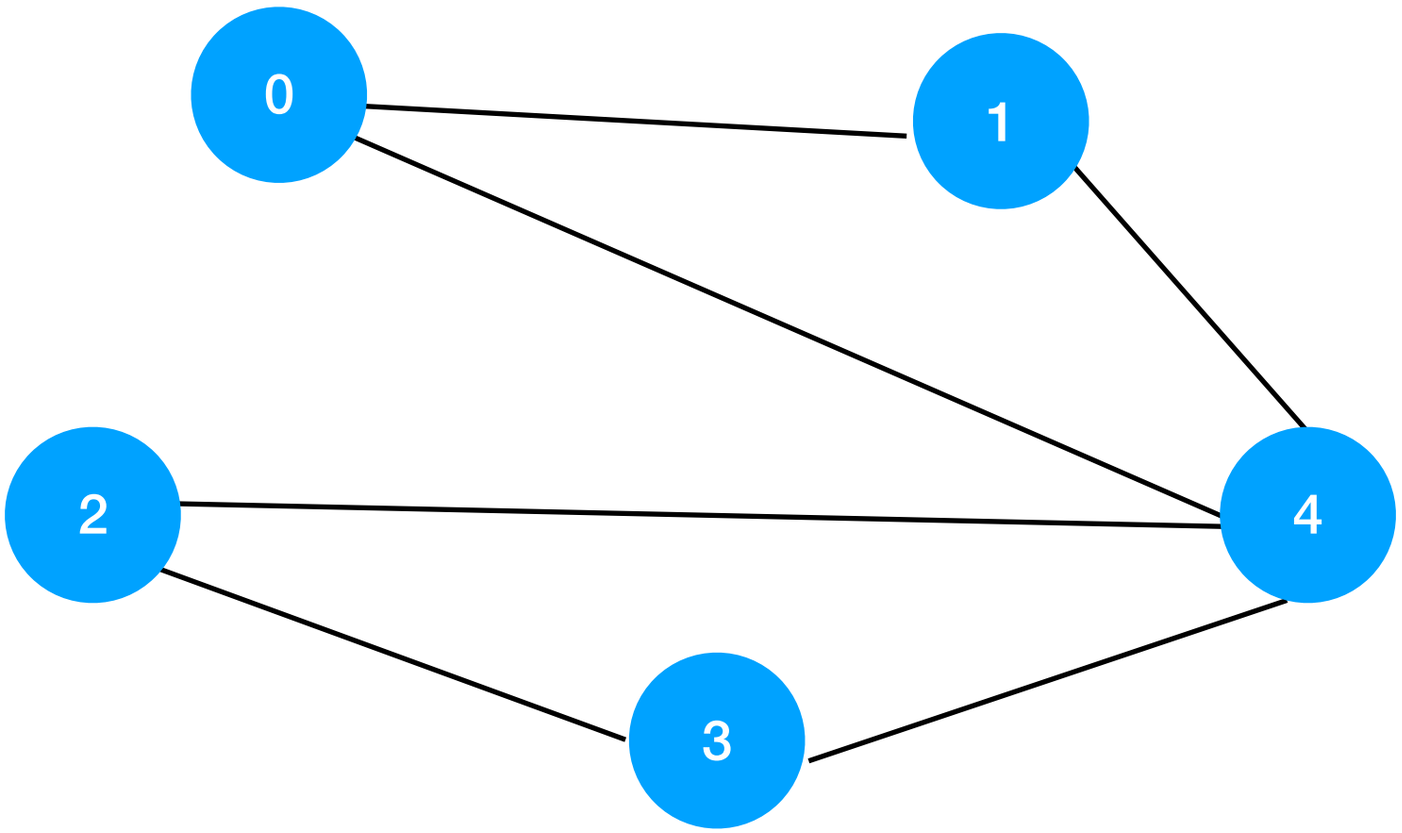
	0	1	2	3	4
0		1			1
1					
2					
3					
4					



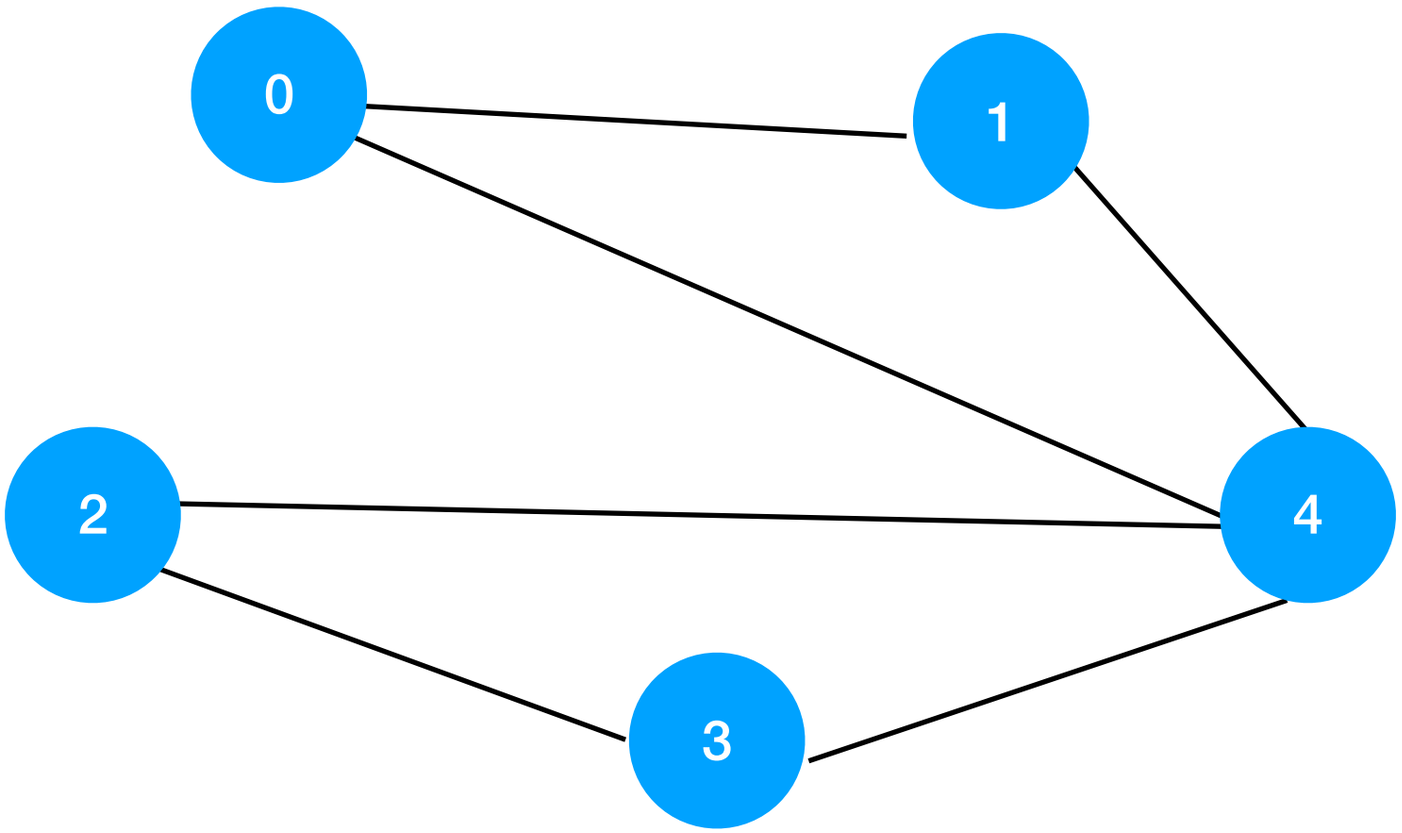
	0	1	2	3	4
0		1			1
1	1				1
2					
3					
4					



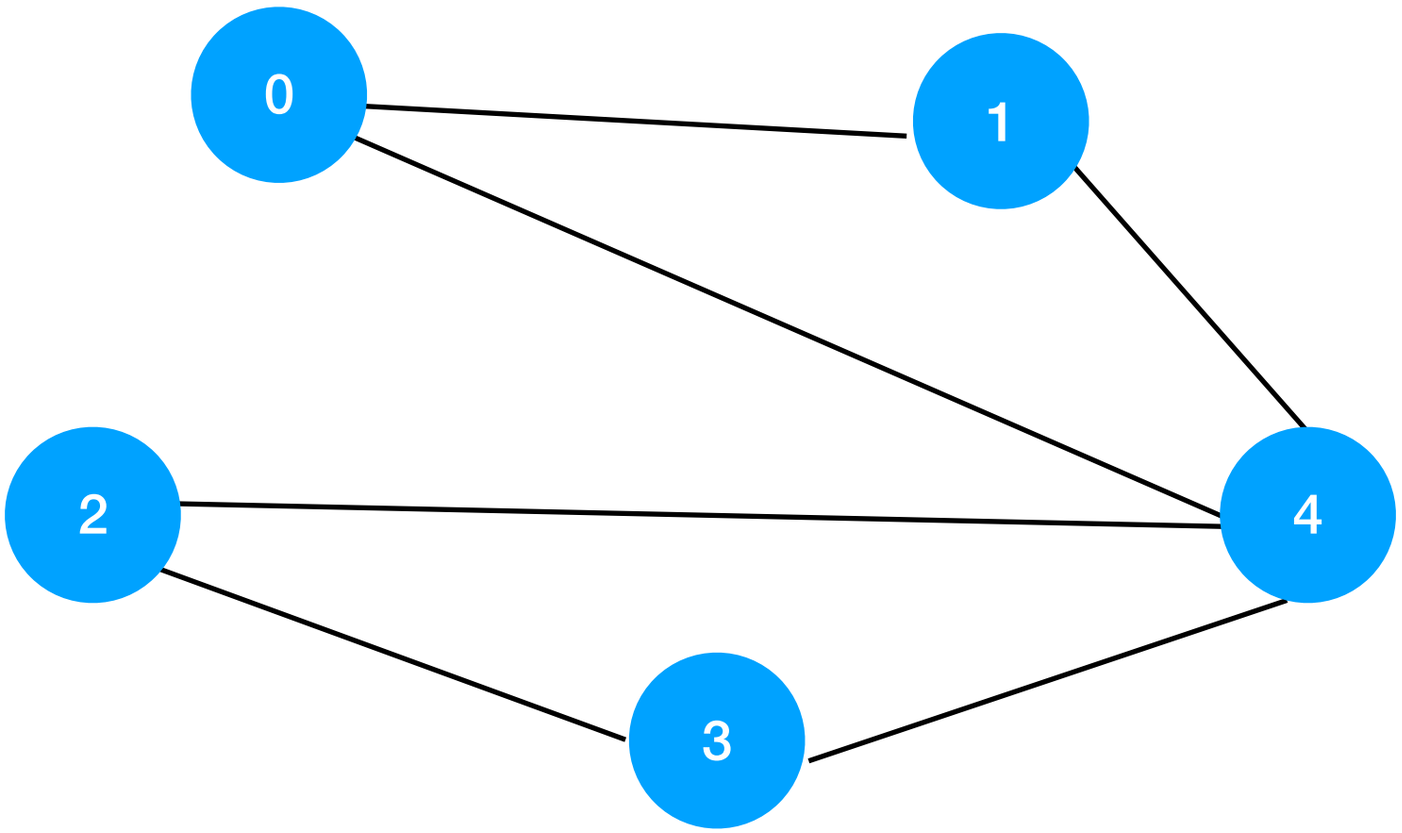
	0	1	2	3	4
0		1			1
1	1				1
2				1	1
3					
4					



	0	1	2	3	4
0		1			1
1	1				1
2				1	1
3			1		1
4					

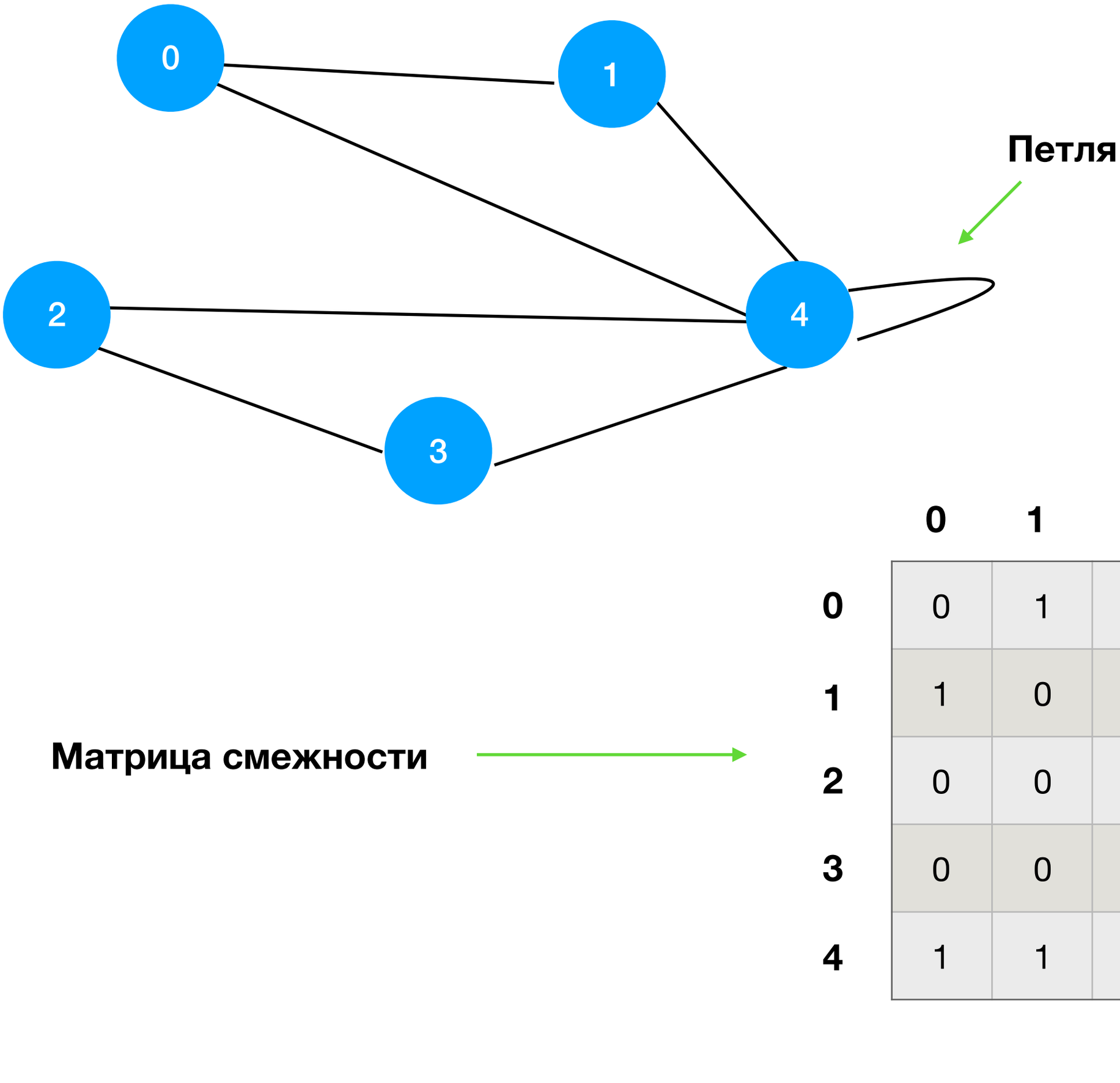


	0	1	2	3	4
0		1			1
1	1				1
2				1	1
3			1		1
4	1	1	1	1	



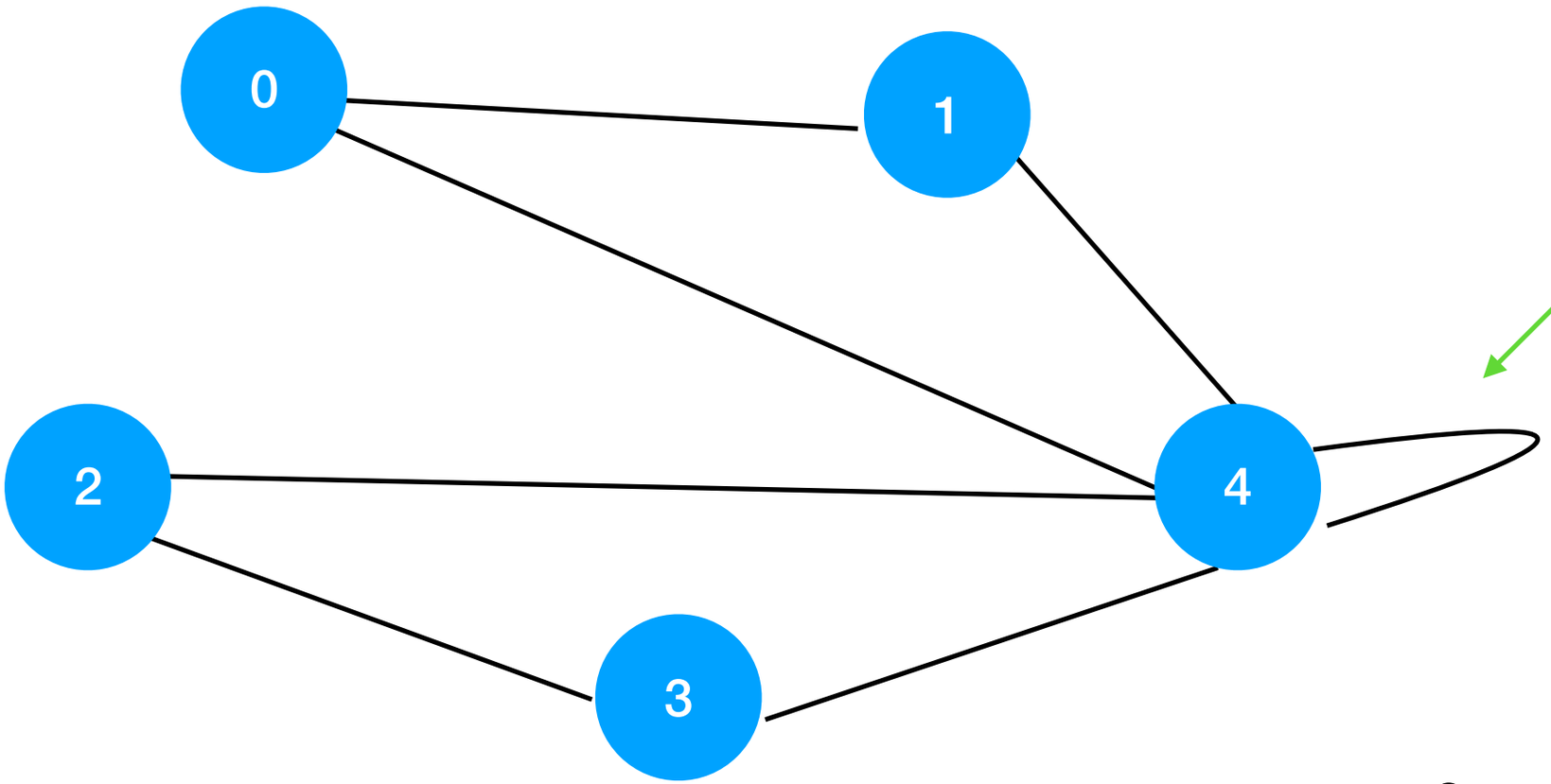
Матрица смежности

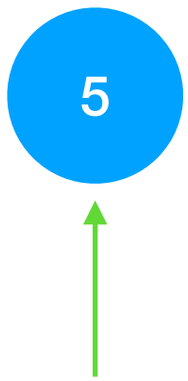
	0	1	2	3	4
0	0	1	0	0	1
1	1	0	0	0	1
2	0	0	0	1	1
3	0	0	1	0	1
4	1	1	1	1	0



Матрица смежности

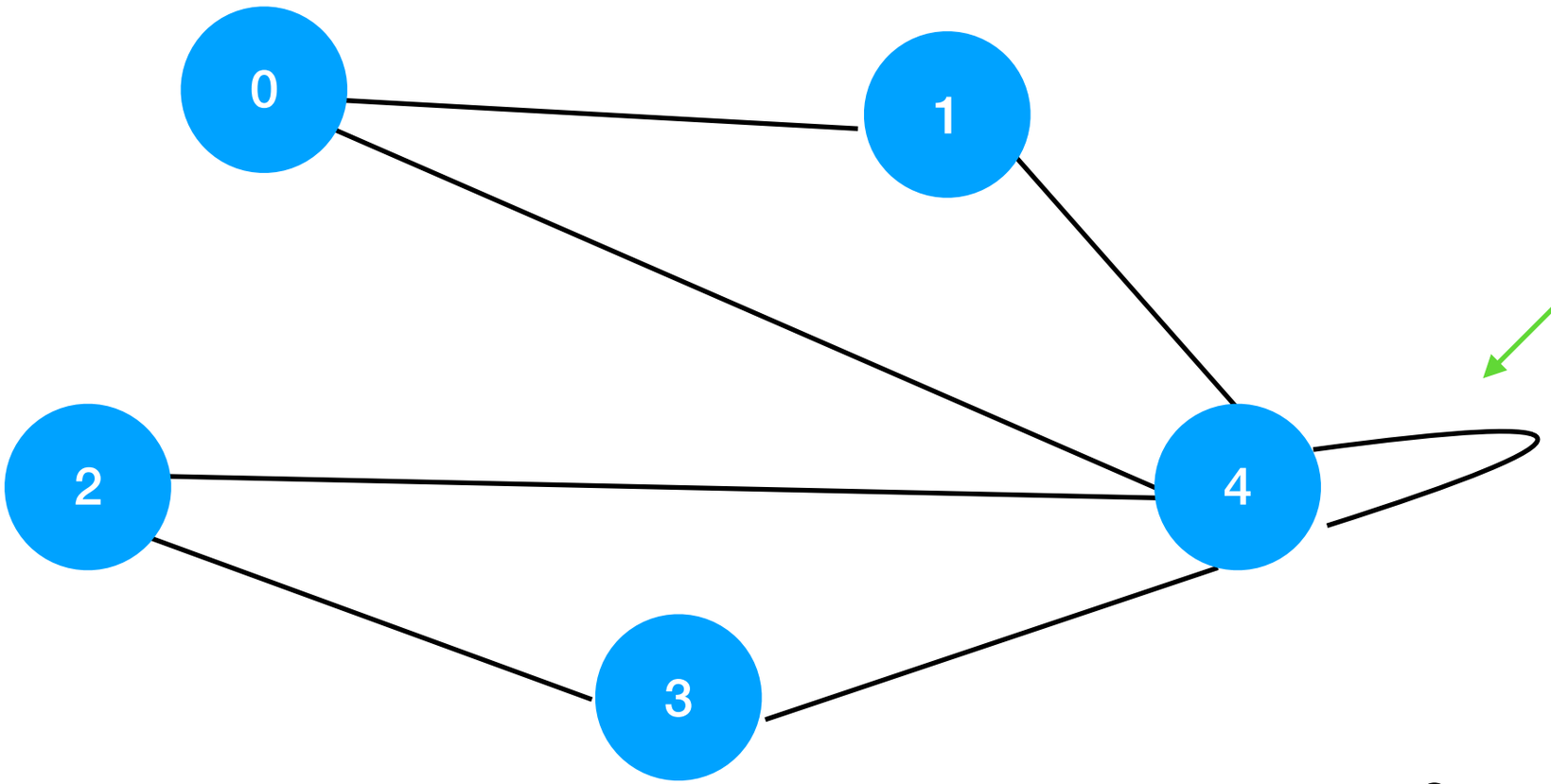
	0	1	2	3	4
0	0	1	0	0	1
1	1	0	0	0	1
2	0	0	0	1	1
3	0	0	1	0	1
4	1	1	1	1	1

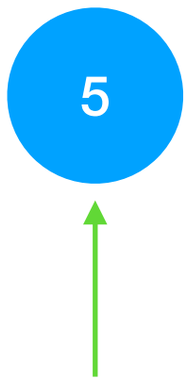




Изолированная вершина

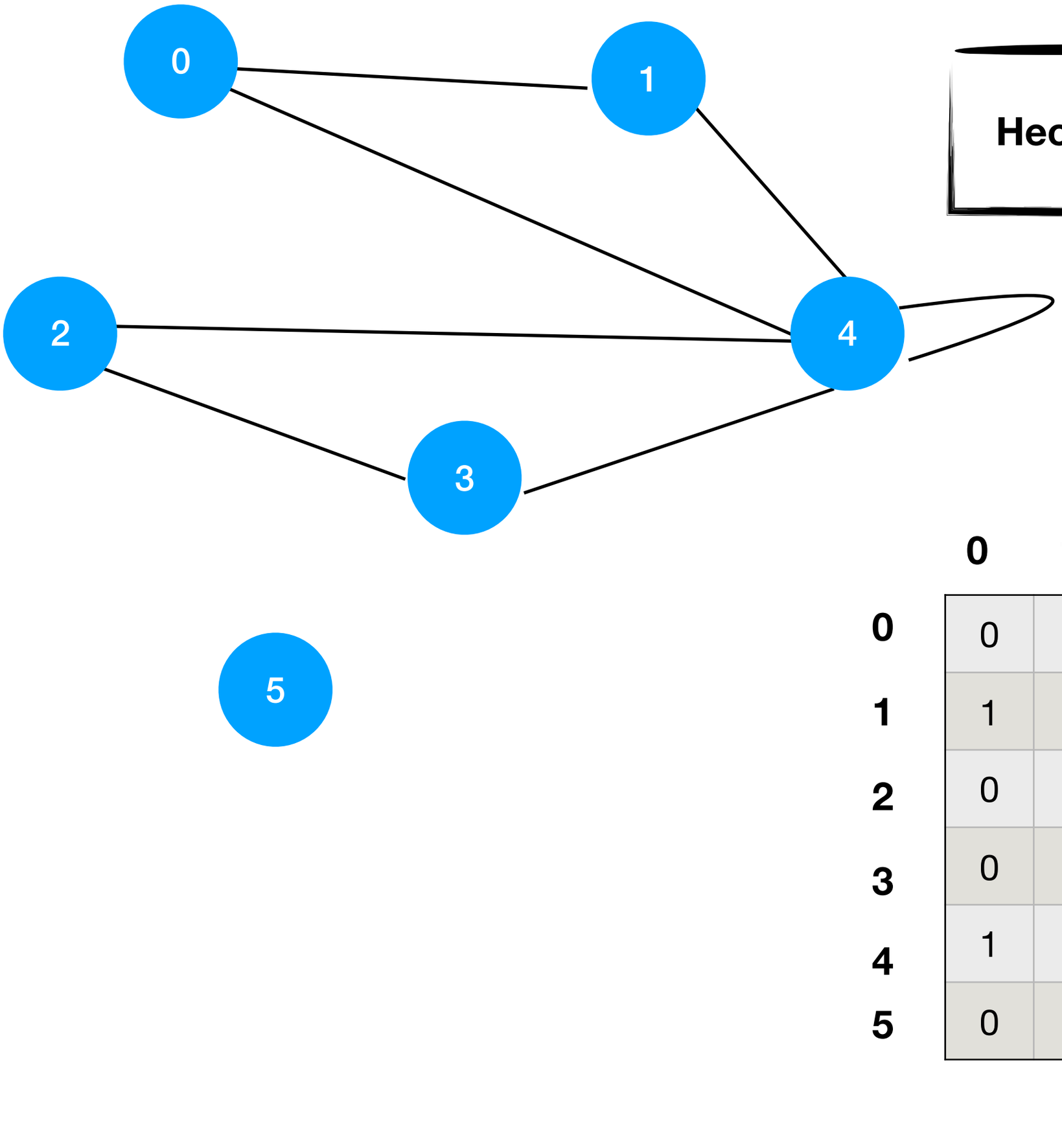
	0	1	2	3	4
0	0	1	0	0	1
1	1	0	0	0	1
2	0	0	0	1	1
3	0	0	1	0	1
4	1	1	1	1	1





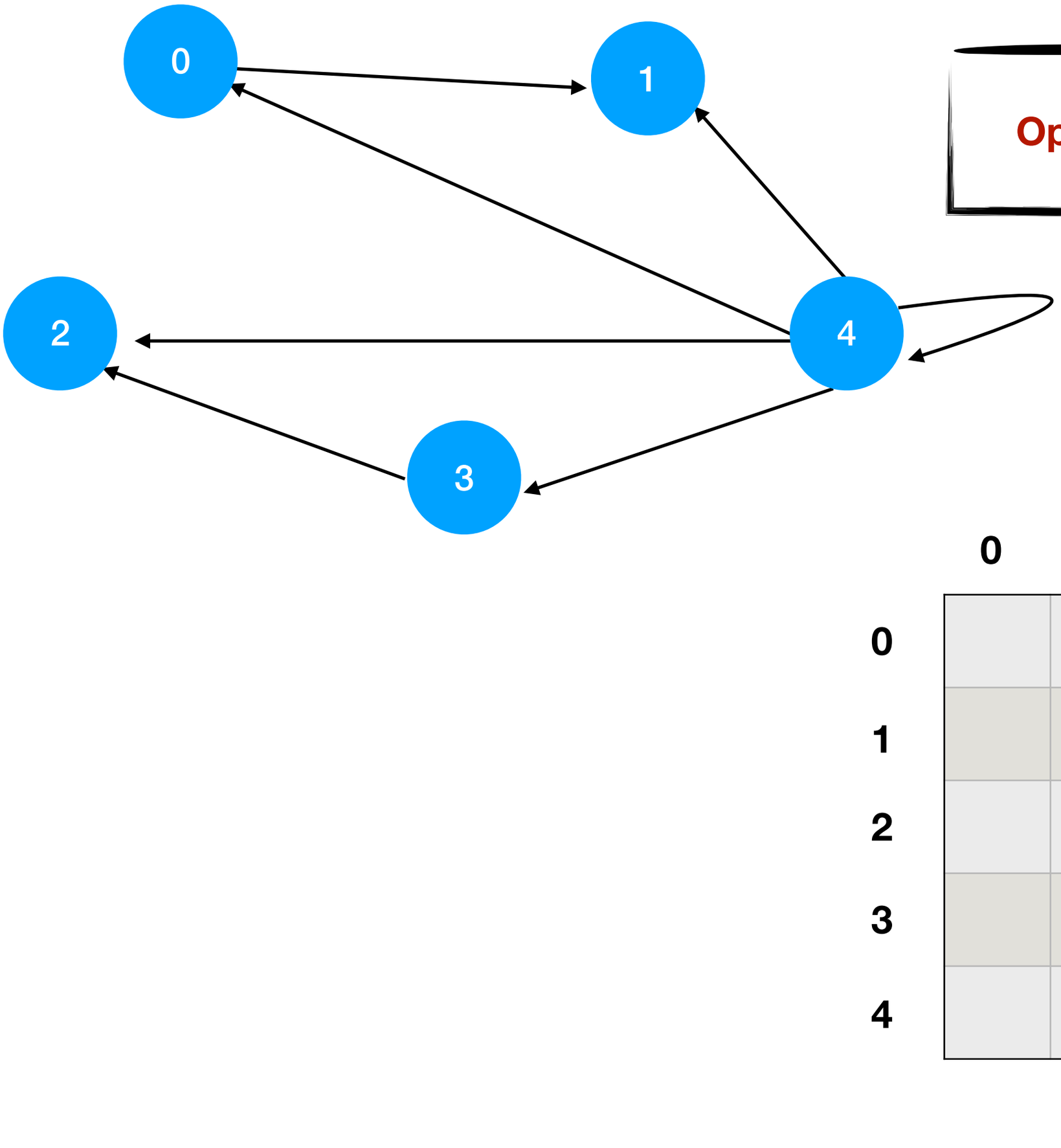
Изолированная вершина

	0	1	2	3	4	5
0	0	1	0	0	1	0
1	1	0	0	0	1	0
2	0	0	0	1	1	0
3	0	0	1	0	1	0
4	1	1	1	1	1	0
5	0	0	0	0	0	0



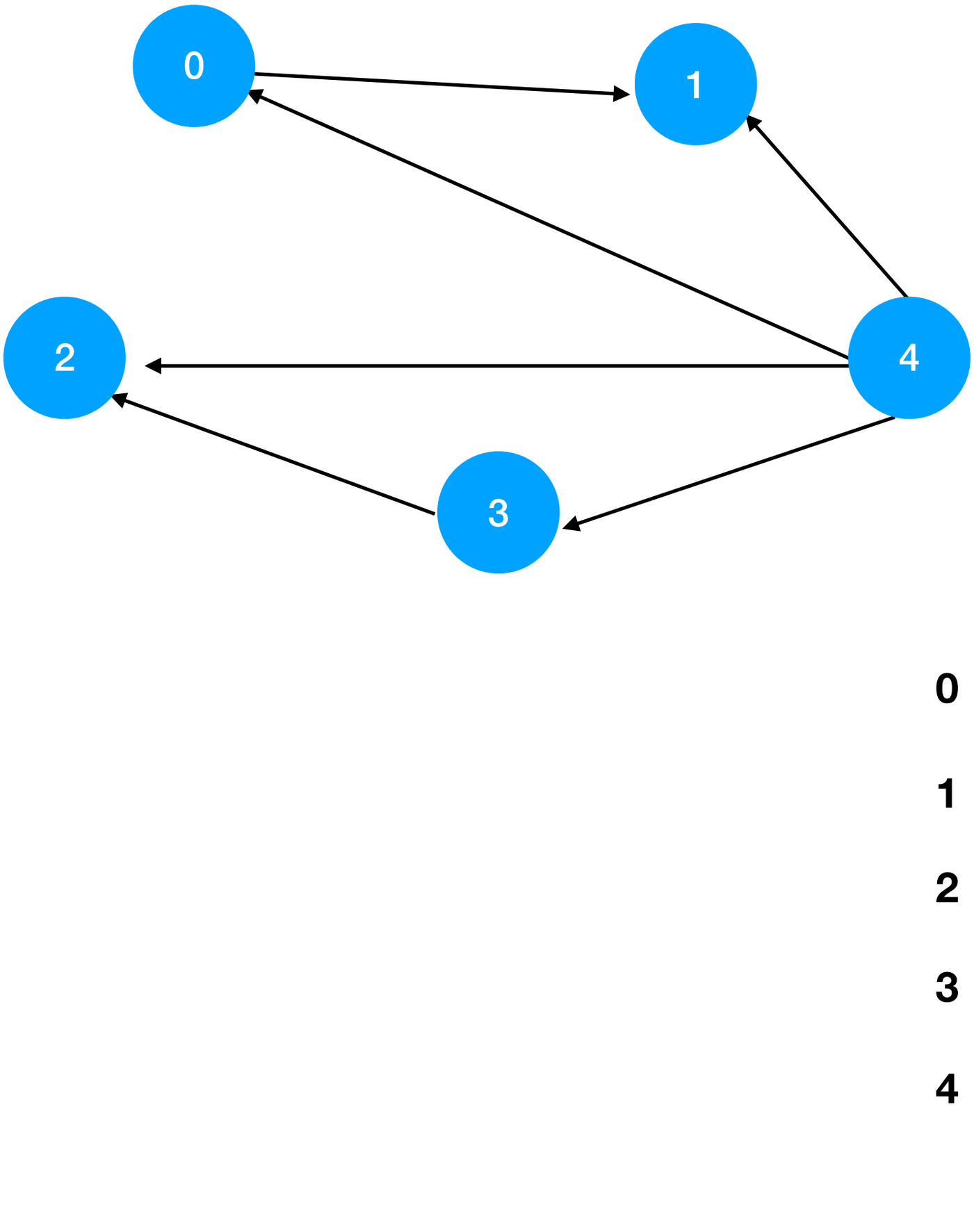
Неориентированный граф

	0	1	2	3	4	5
0	0	1	0	0	1	0
1	1	0	0	0	1	0
2	0	0	0	1	1	0
3	0	0	1	0	1	0
4	1	1	1	1	1	0
5	0	0	0	0	0	0



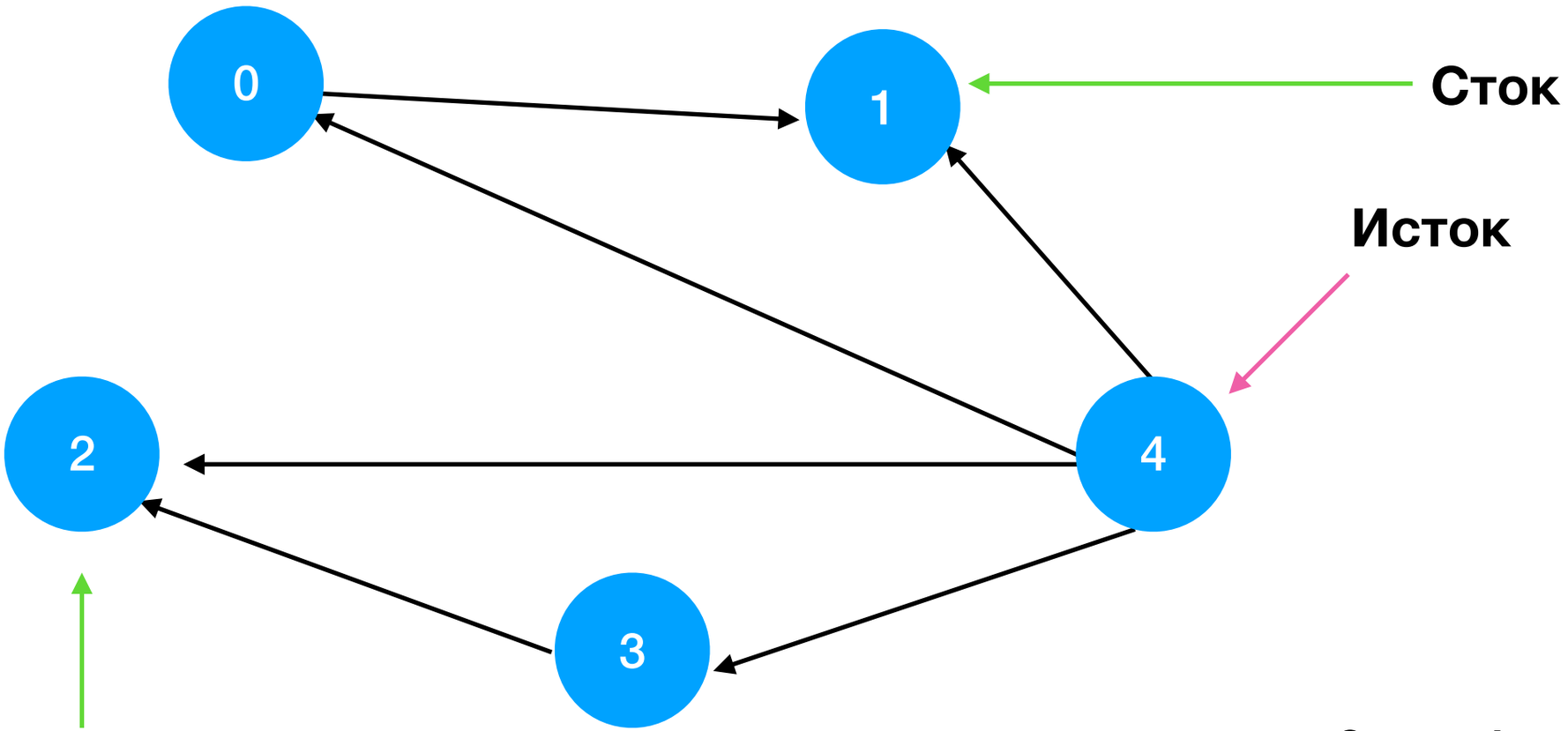
Ориентированный граф

	0	1	2	3	4
0					
1					
2					
3					
4					



Ориентированный граф

	0	1	2	3	4
0		1			
1					
2					
3			1		
4	1	1	1	1	

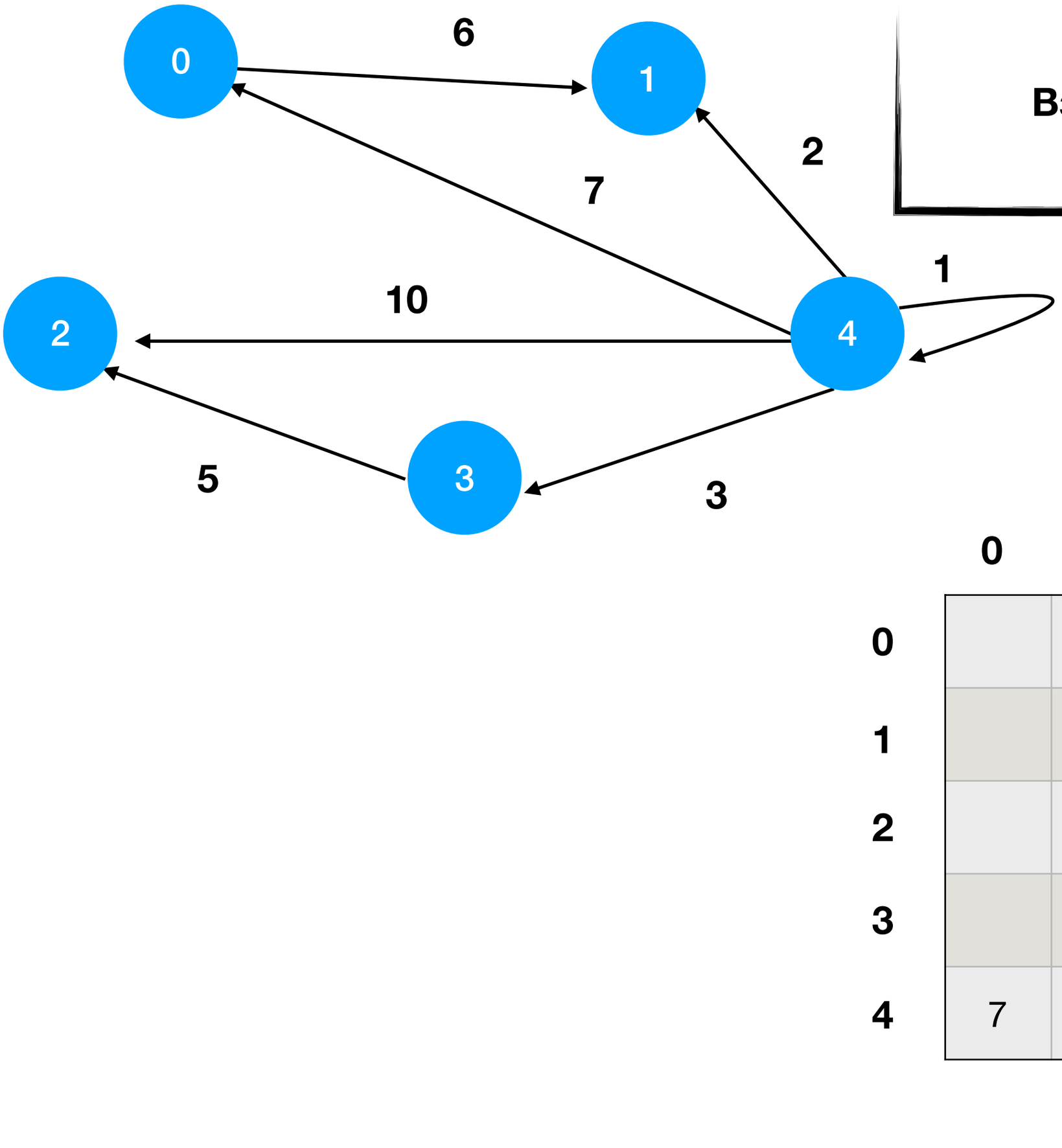


Сток

Сток - не выходит ни одного ребра

Исток - не входит ни одного ребра

	0	1	2	3	4
0		1			
1					
2					
3			1		
4	1	1	1	1	



Взвешенный граф

	0	1	2	3	4
0		6			
1					
2					
3			5		
4	7	2	10	3	1

	0	1	2	3	4
0	0	6	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	5	0	0
4	7	2	10	3	1

Двумерный массив

	столбец 0	столбец 1	столбец 2	столбец 3
строка 0	**arr[0][0]**	**arr[0][1]**	**arr[0][2]**	**arr[0][3]**
строка 1	**arr[1][0]**	**arr[1][1]**	**arr[1][2]**	**arr[1][3]**
строка 2	**arr[2][0]**	**arr[2][1]**	**arr[2][2]**	**arr[2][3]**

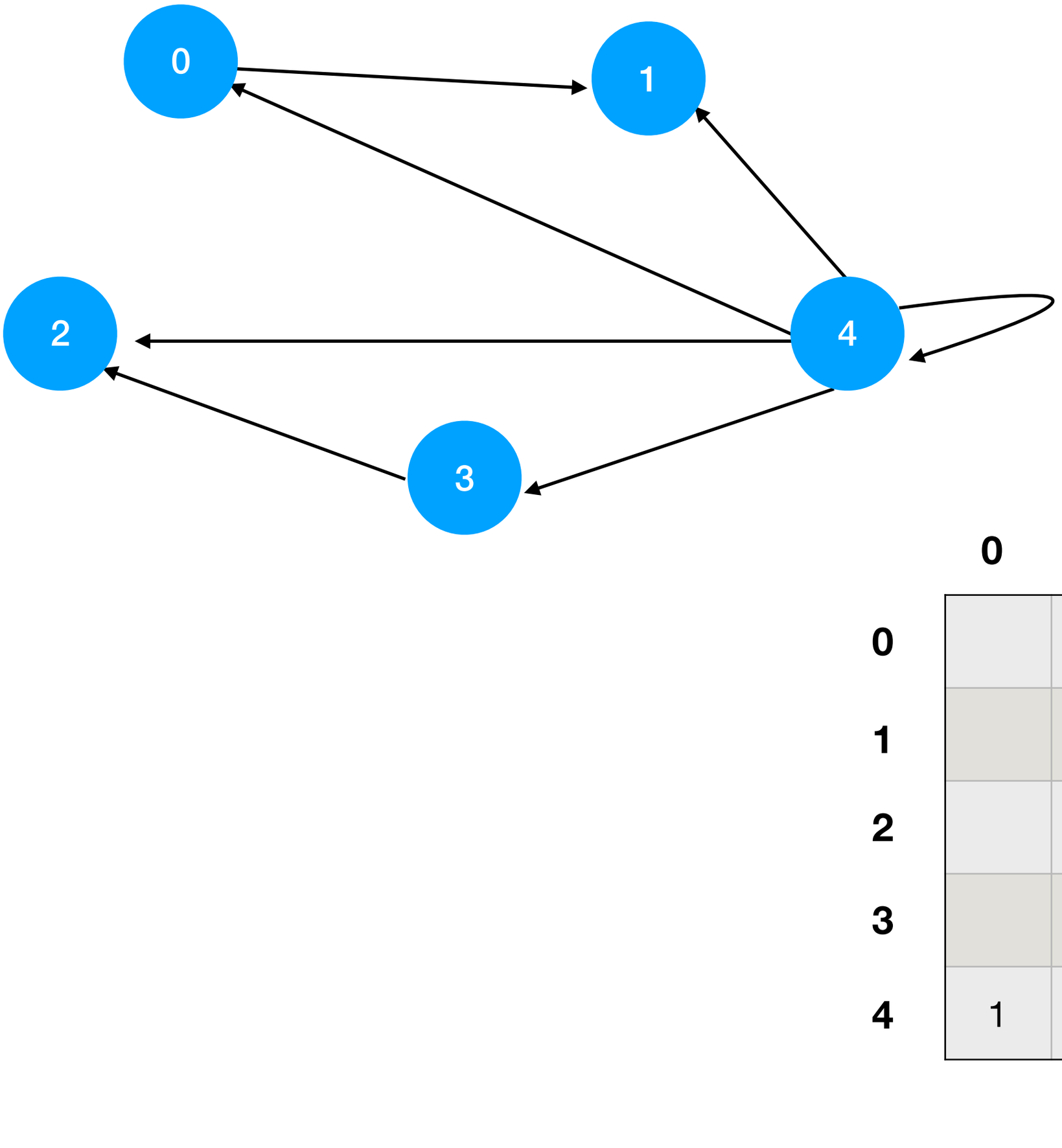
	0	1	2	3	4
0	0	6	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	5	0	0
4	7	2	10	3	1

Java

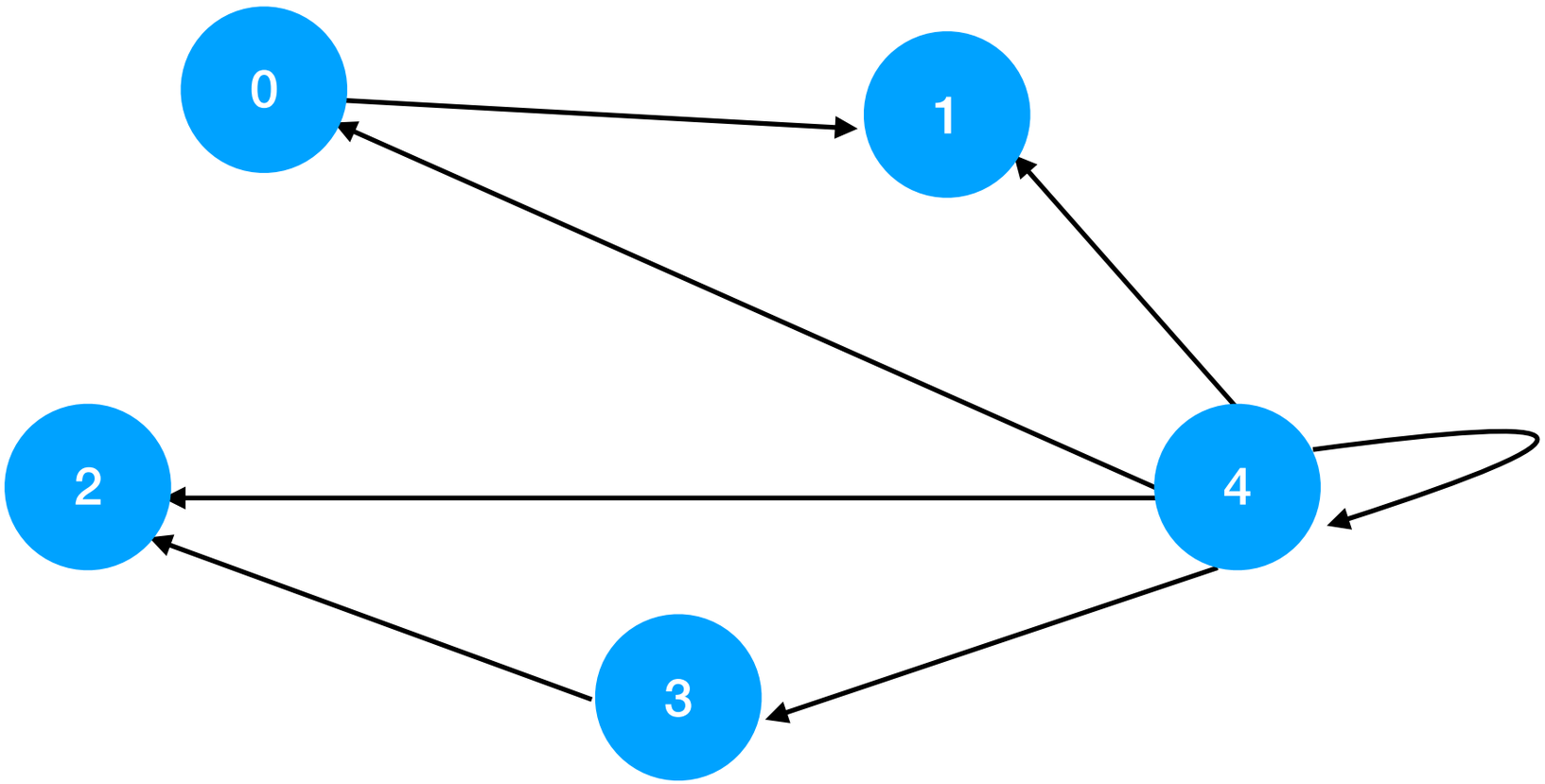
```java
int[][] a = new int[N][N];
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                a[i][j] = 0;
            }
        }
```

C++

```cpp
int arr[n][n];
    for(int i = 0; i < n; i++)
        for(int j = 0; j < n; j++){
            arr[i][j] = 0;
        }
```



	0	1	2	3	4
0		1			
1					
2					
3			1		
4	1	1	1	1	1



Список смежности

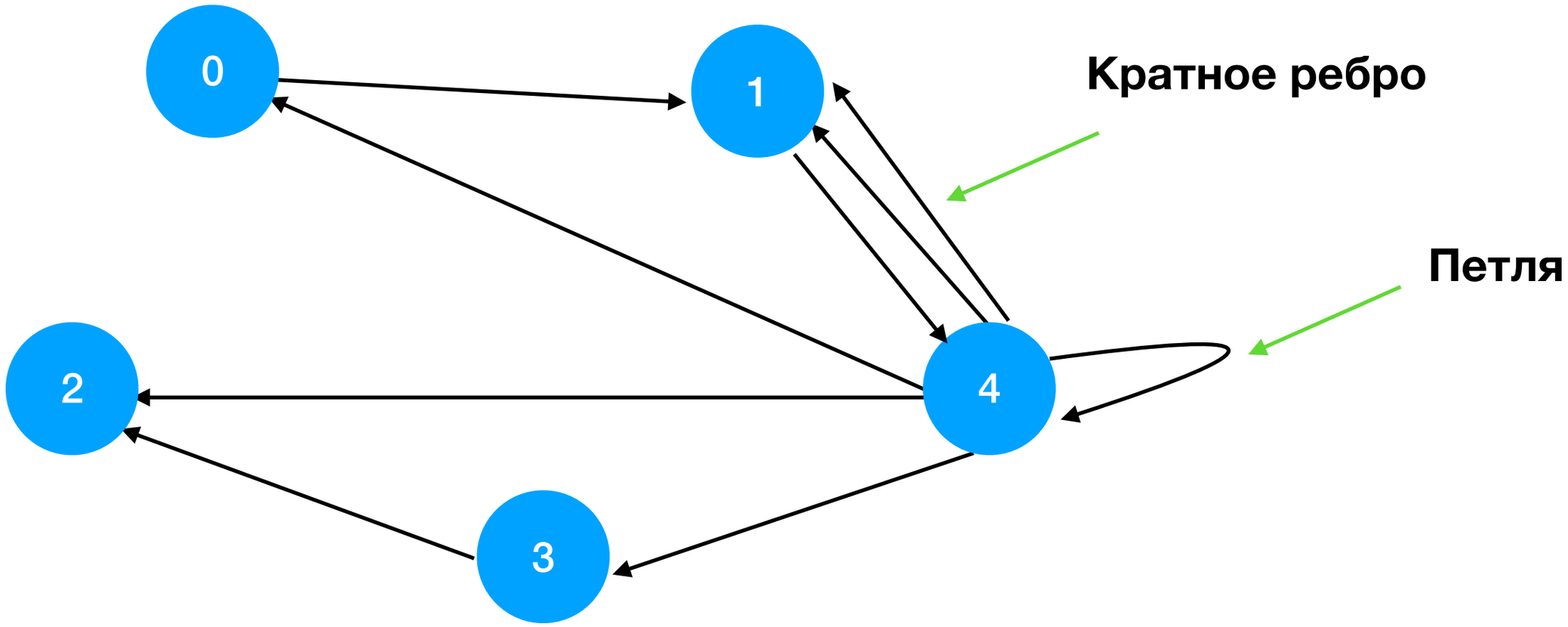
0: 1

1:

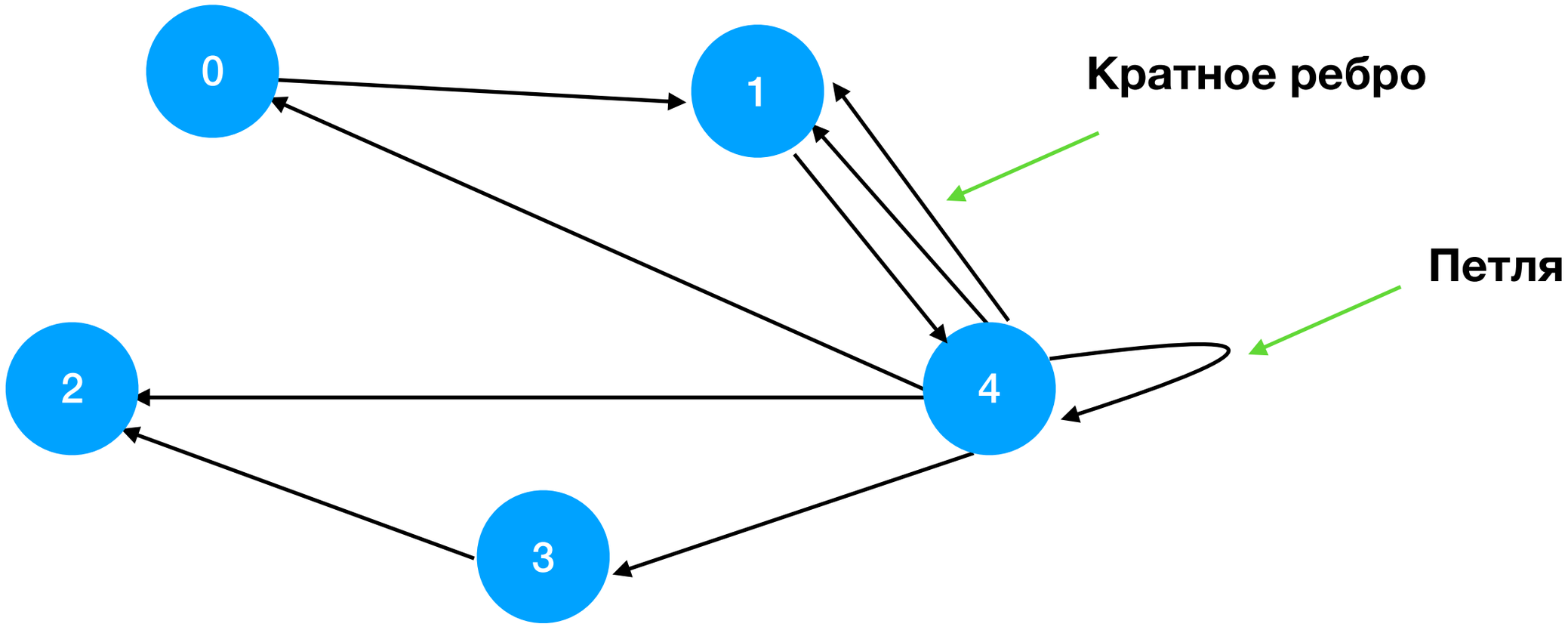
2:

3: 2

4: 0, 1, 2, 3

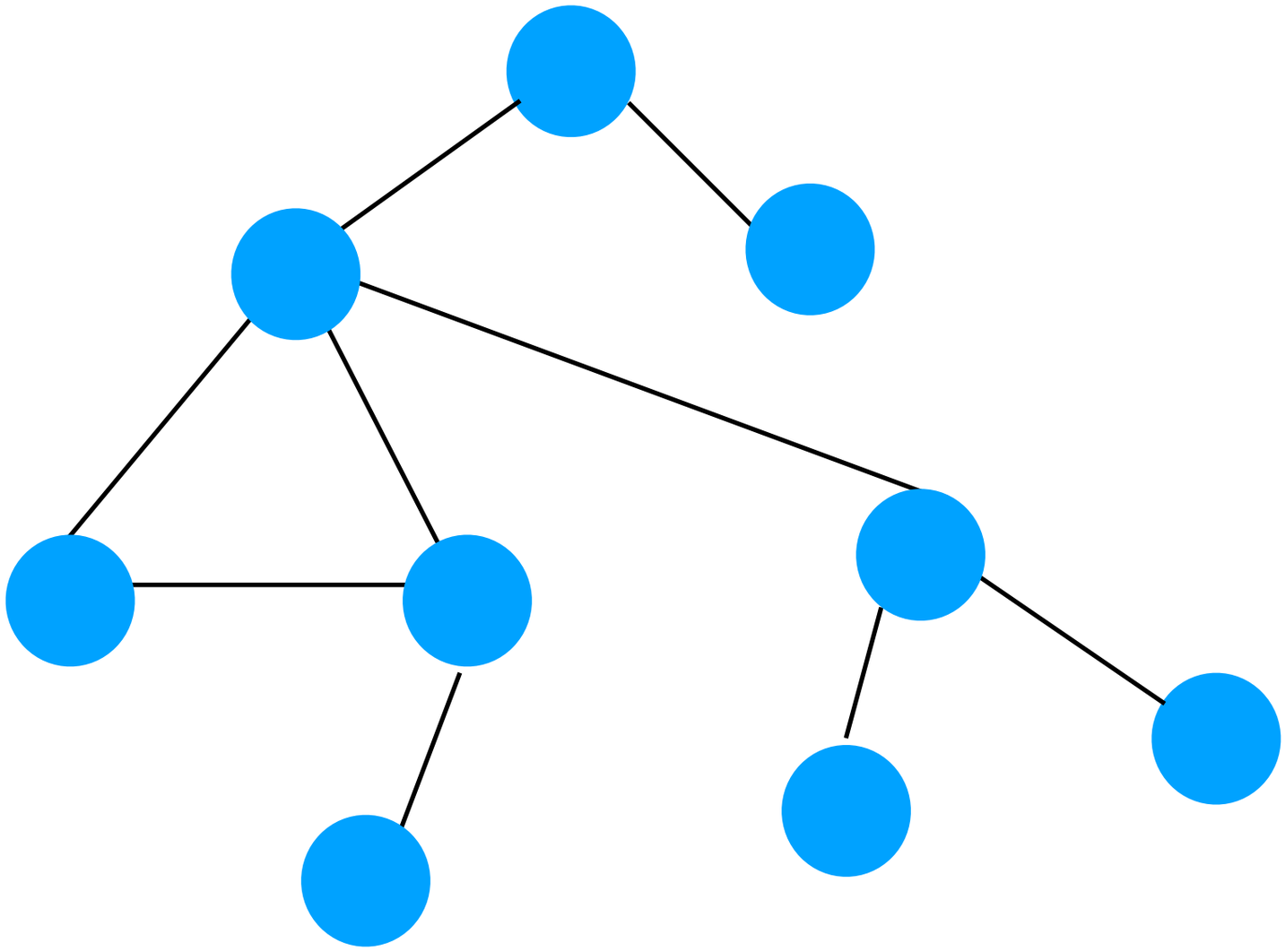


Простой граф - граф, в котором нет петель и кратных ребер

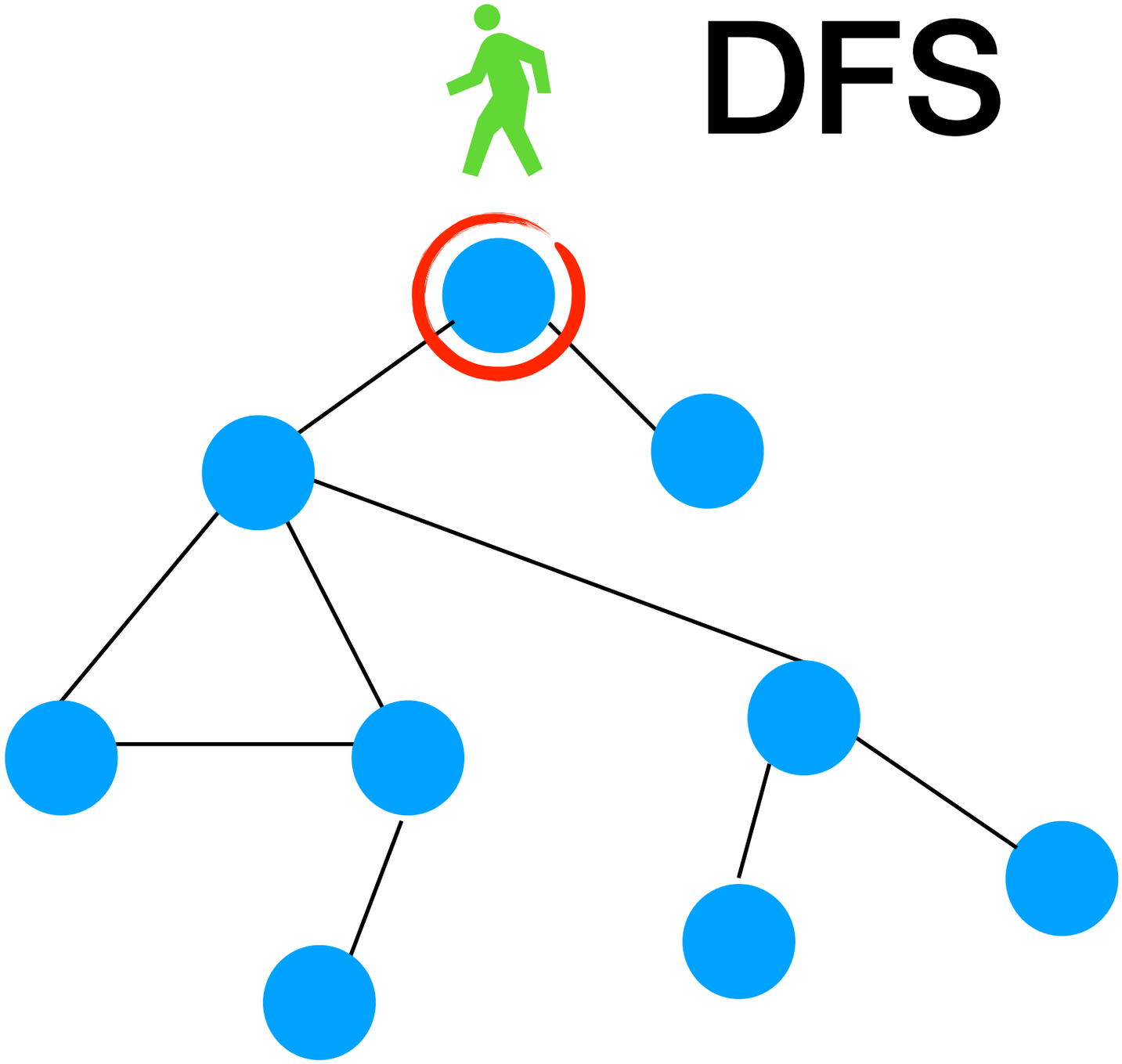


Простой граф - граф, в котором нет петель и кратных ребер

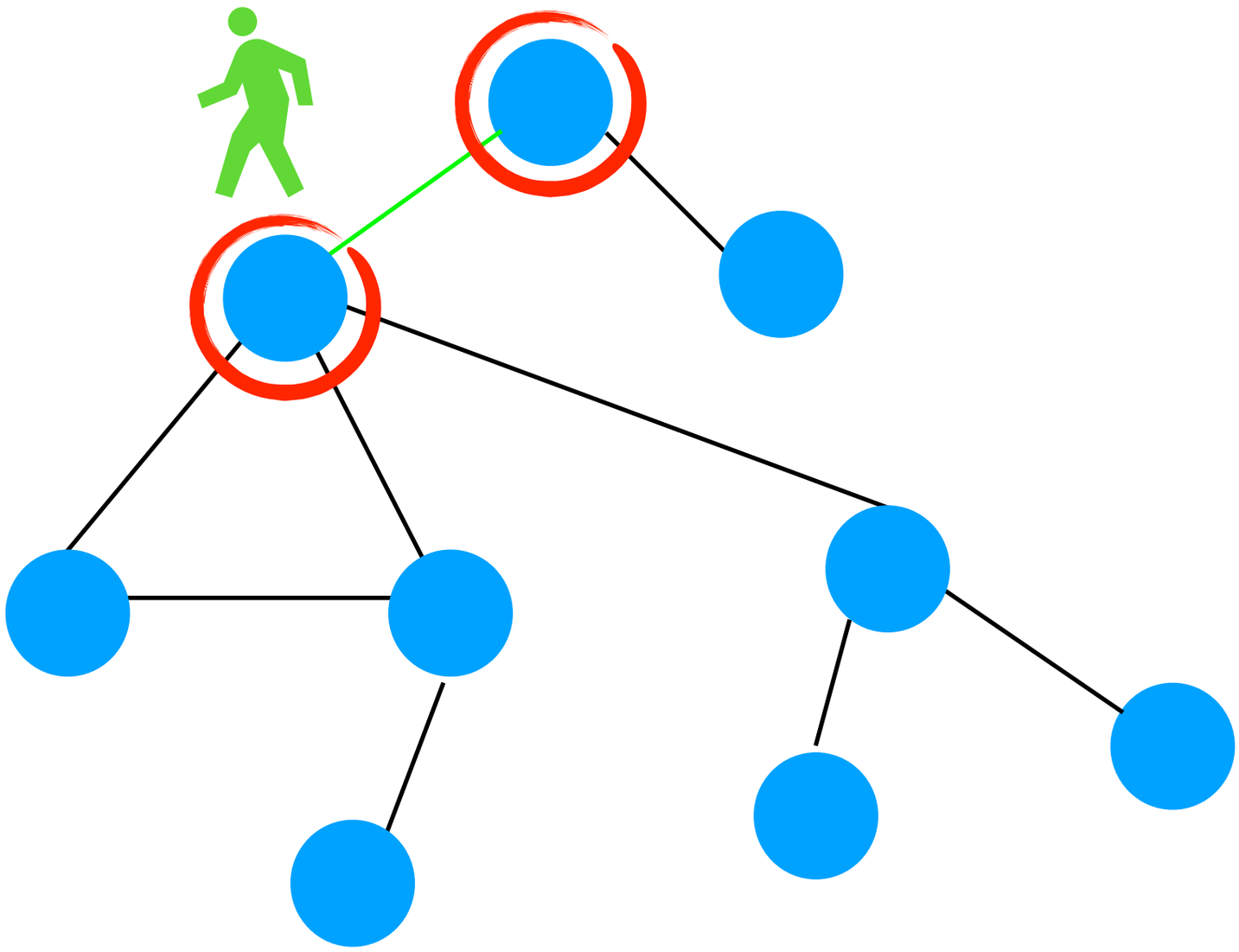
DFS



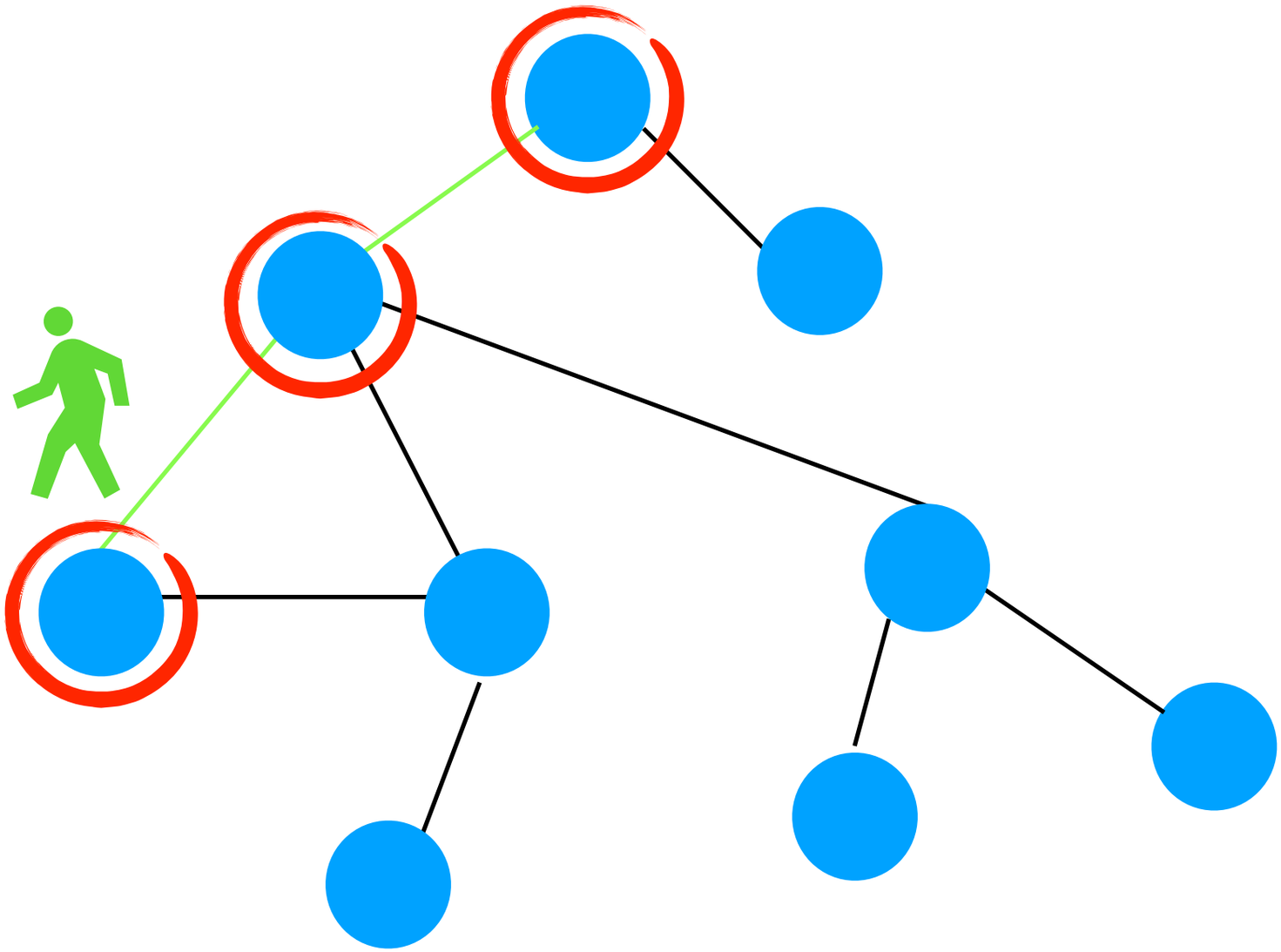
DFS



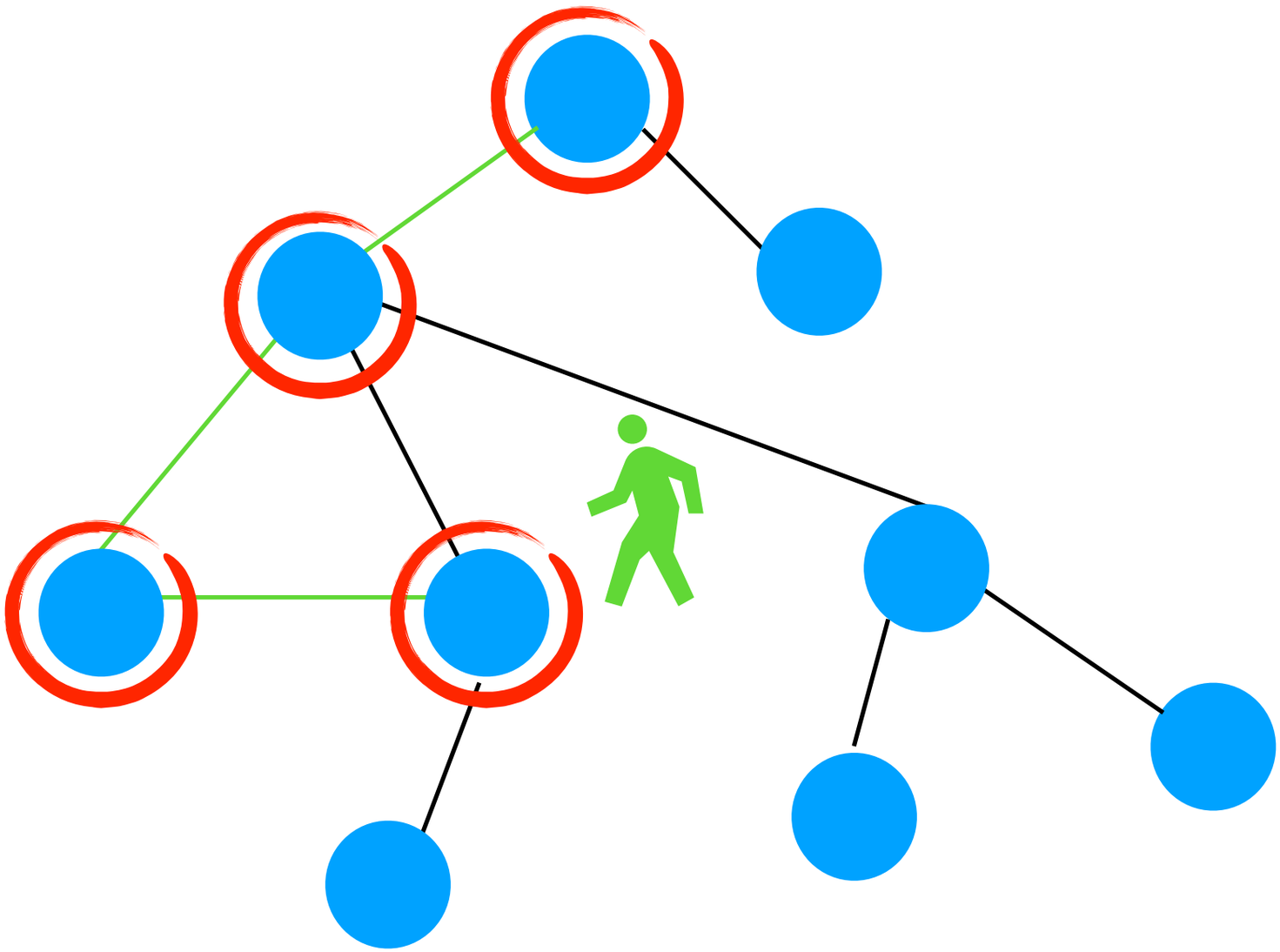
DFS



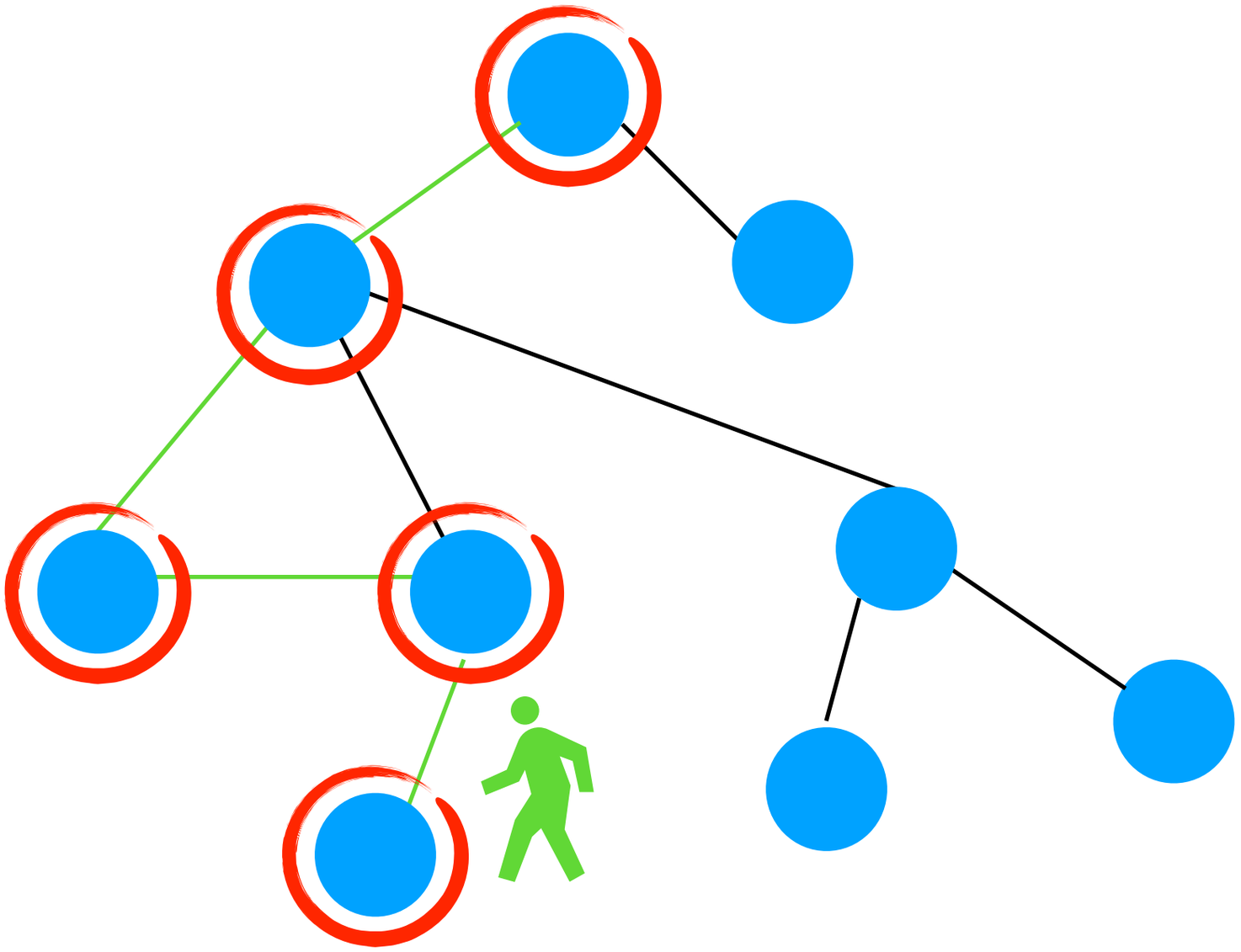
DFS



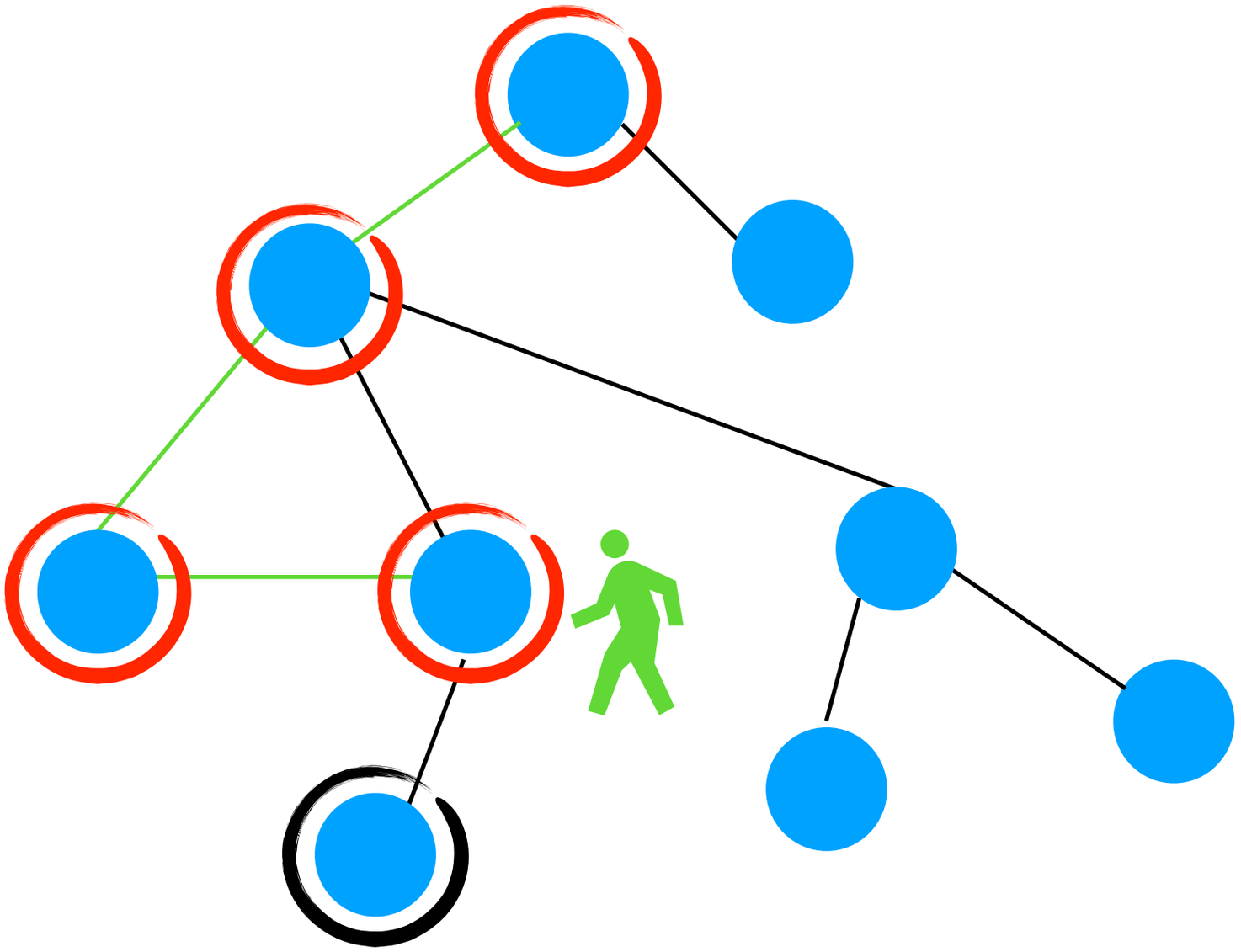
DFS



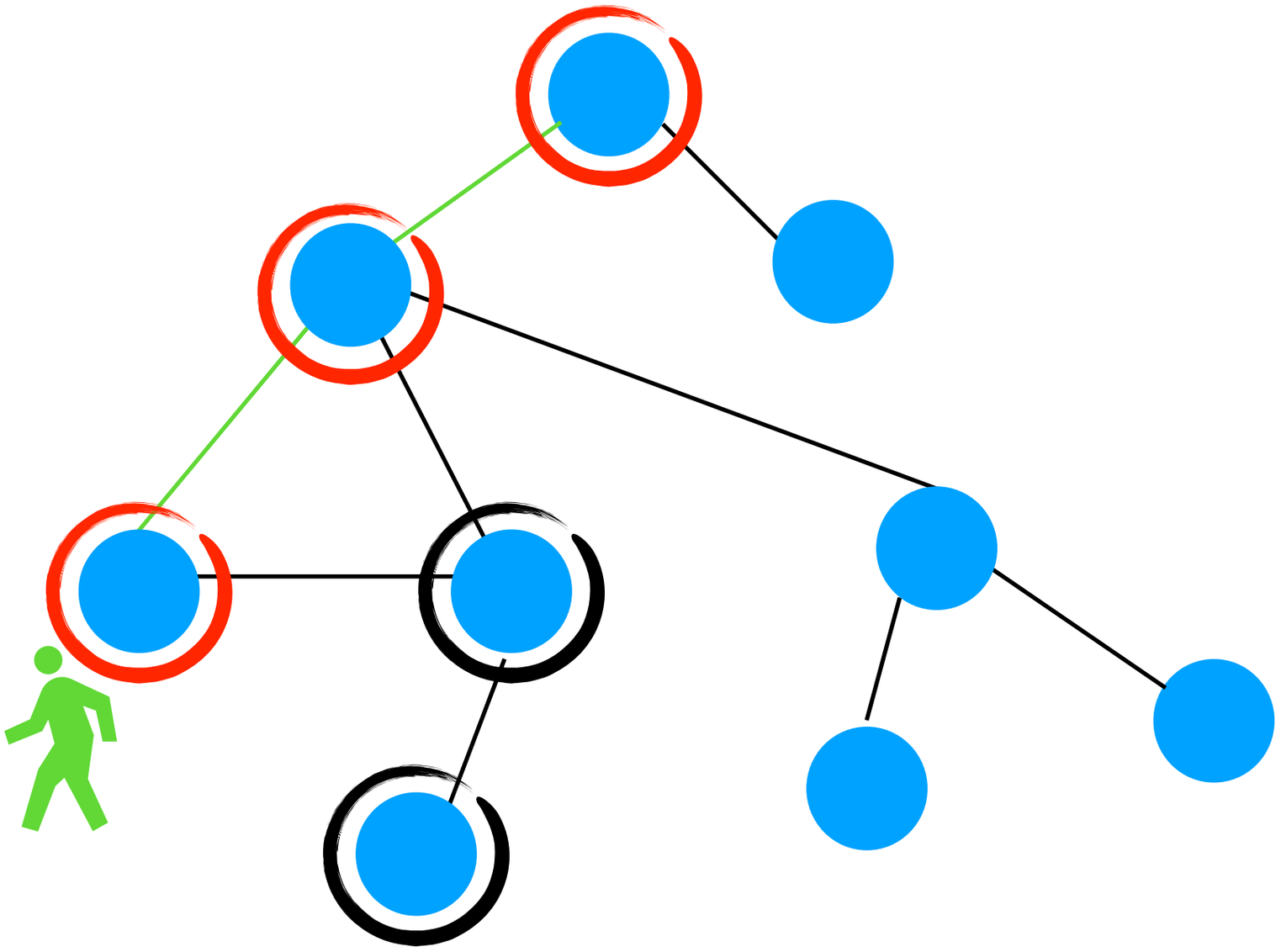
DFS



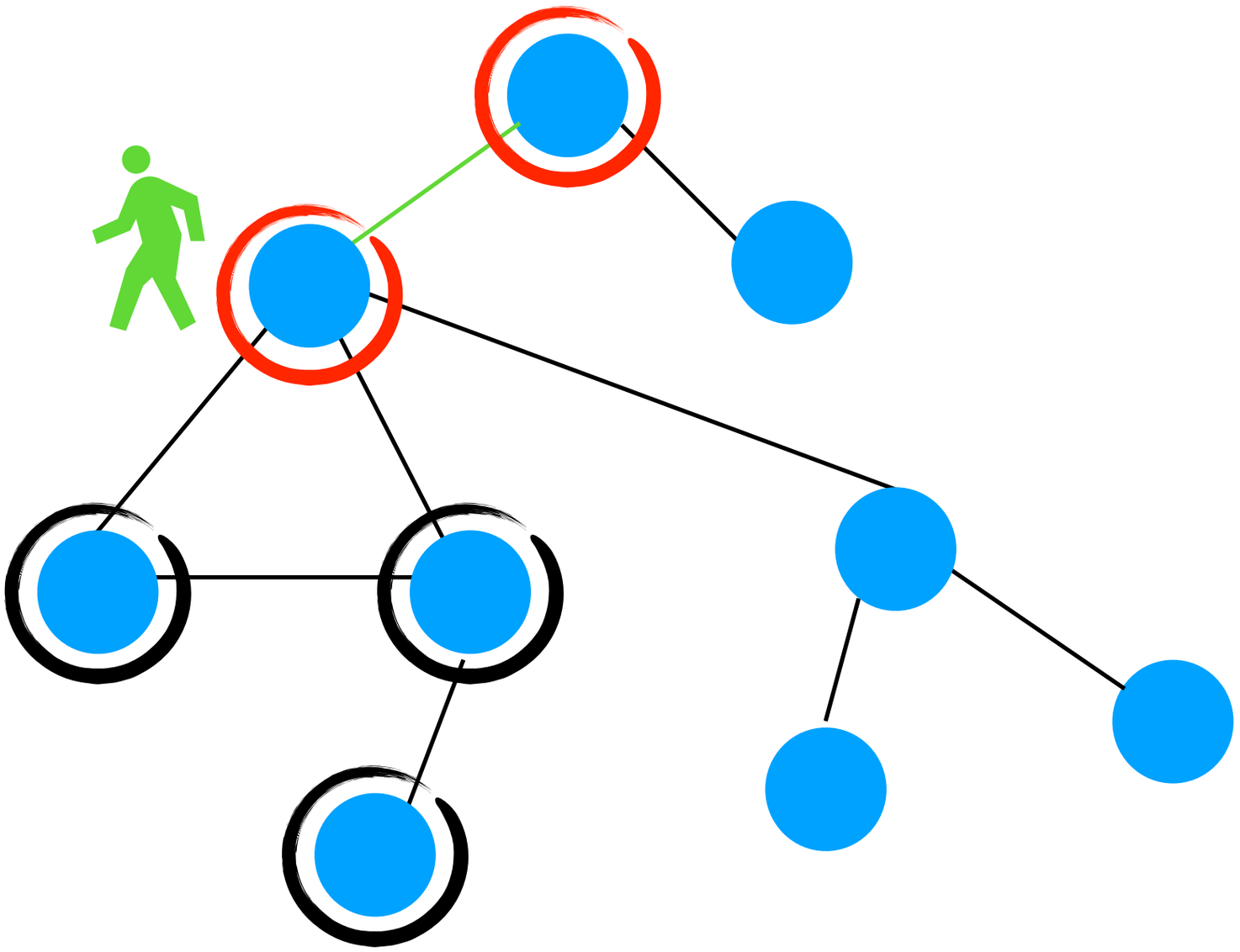
DFS



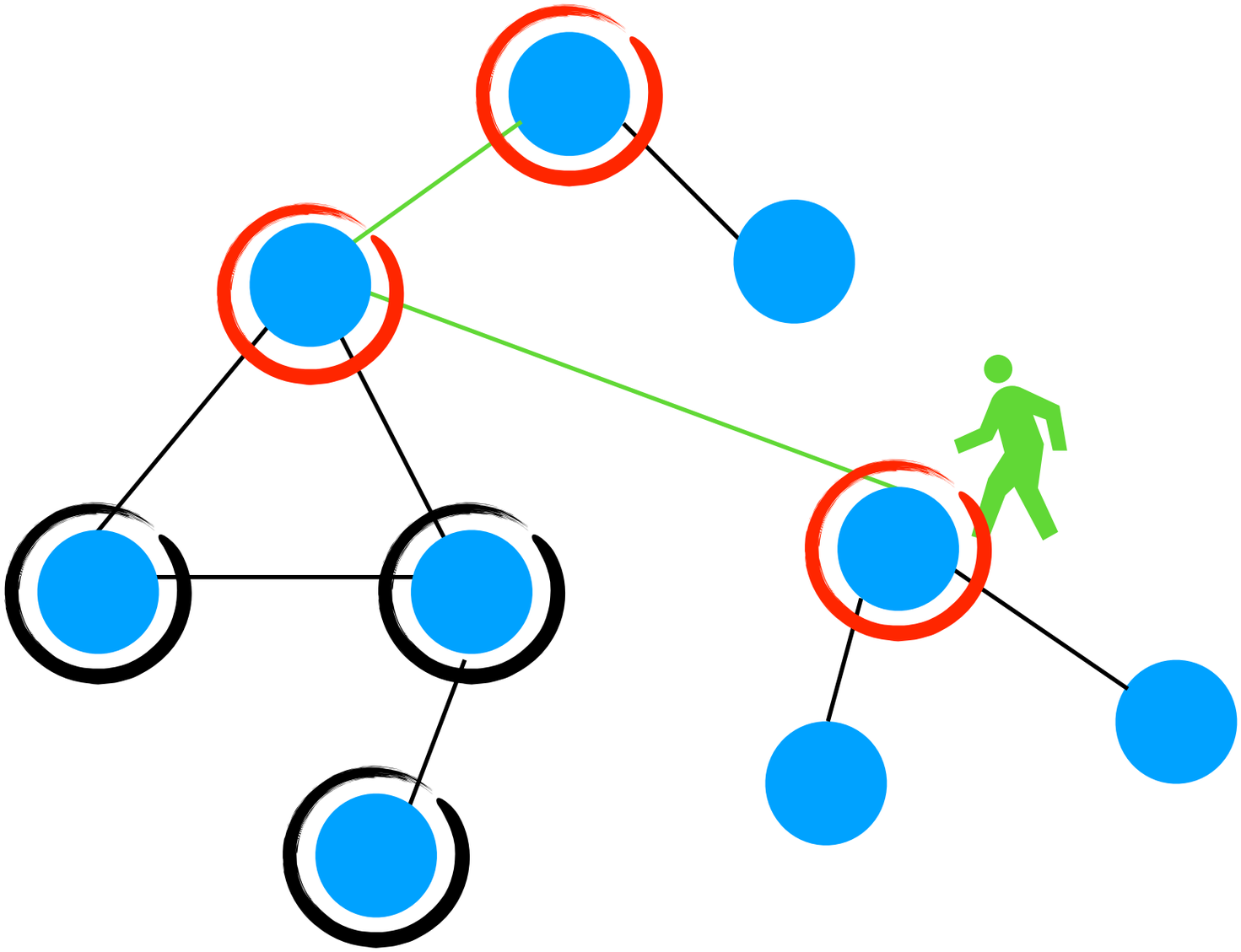
DFS



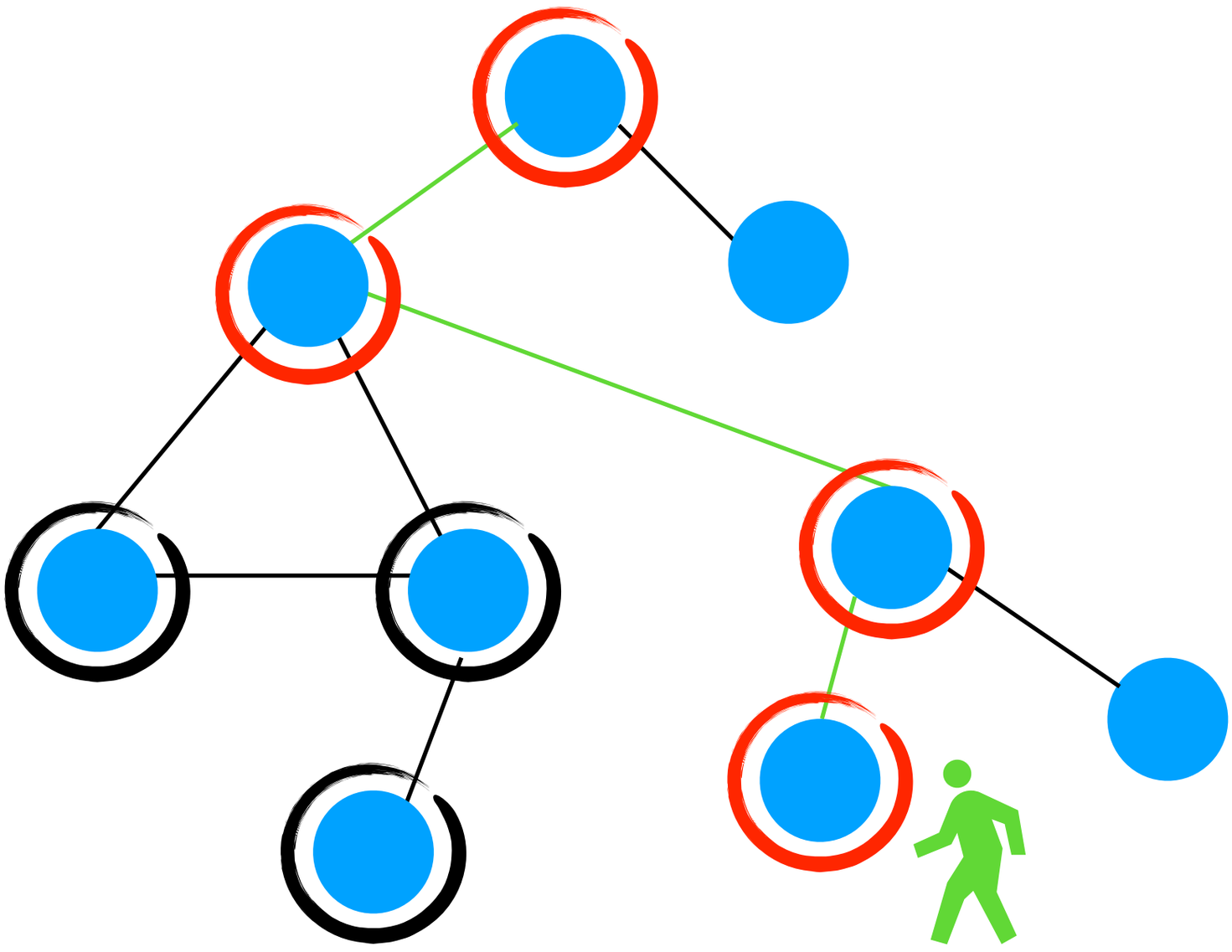
DFS



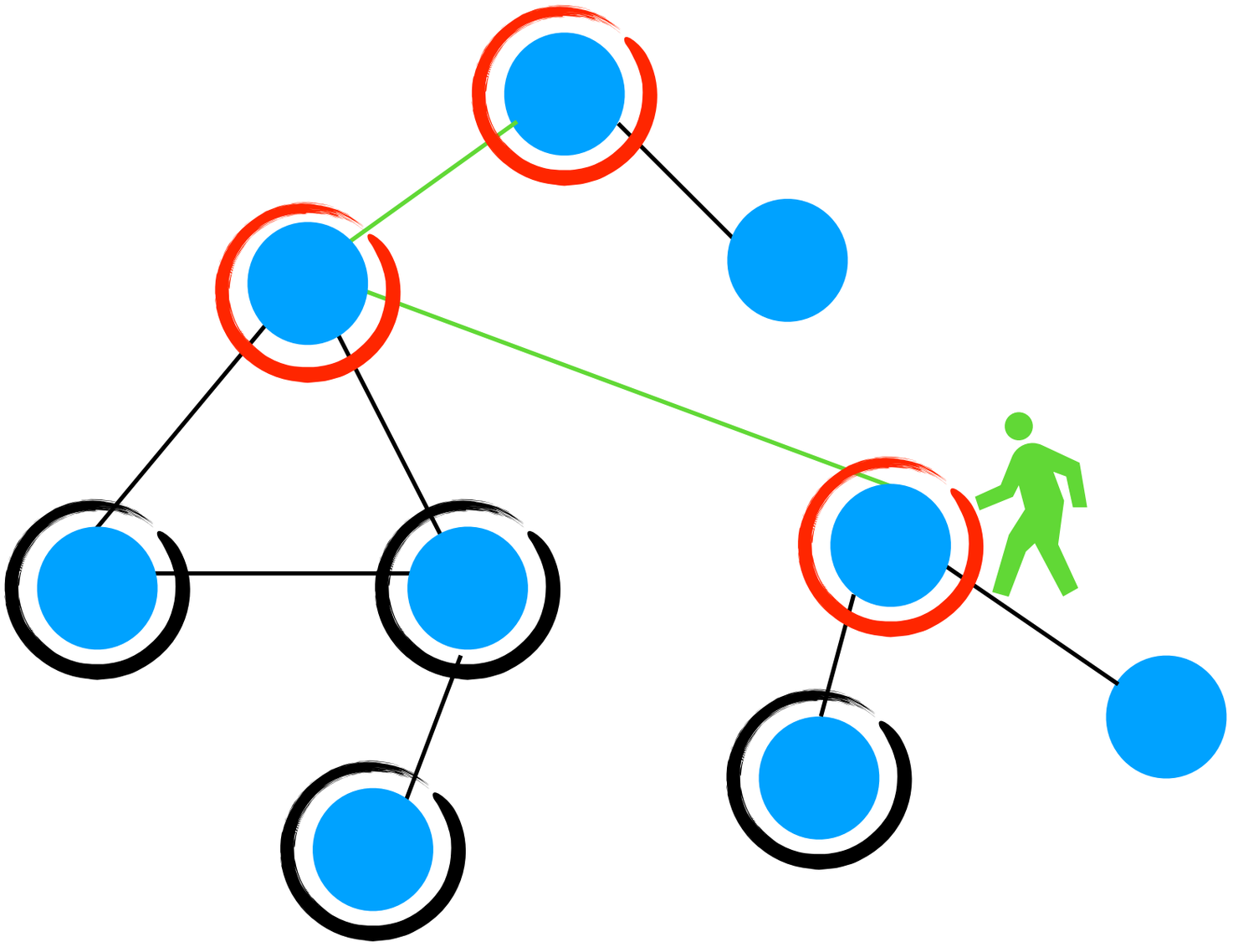
DFS



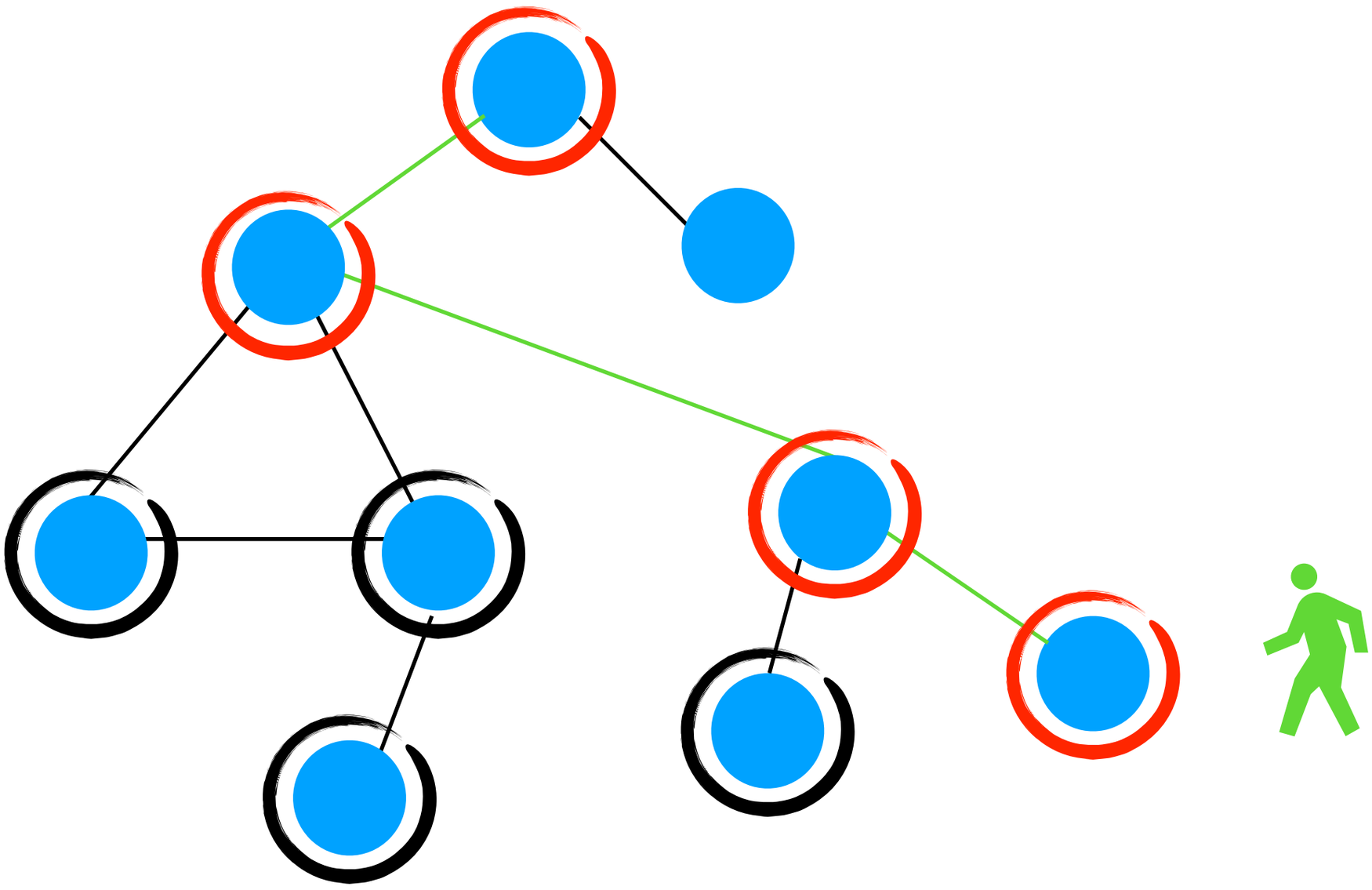
DFS



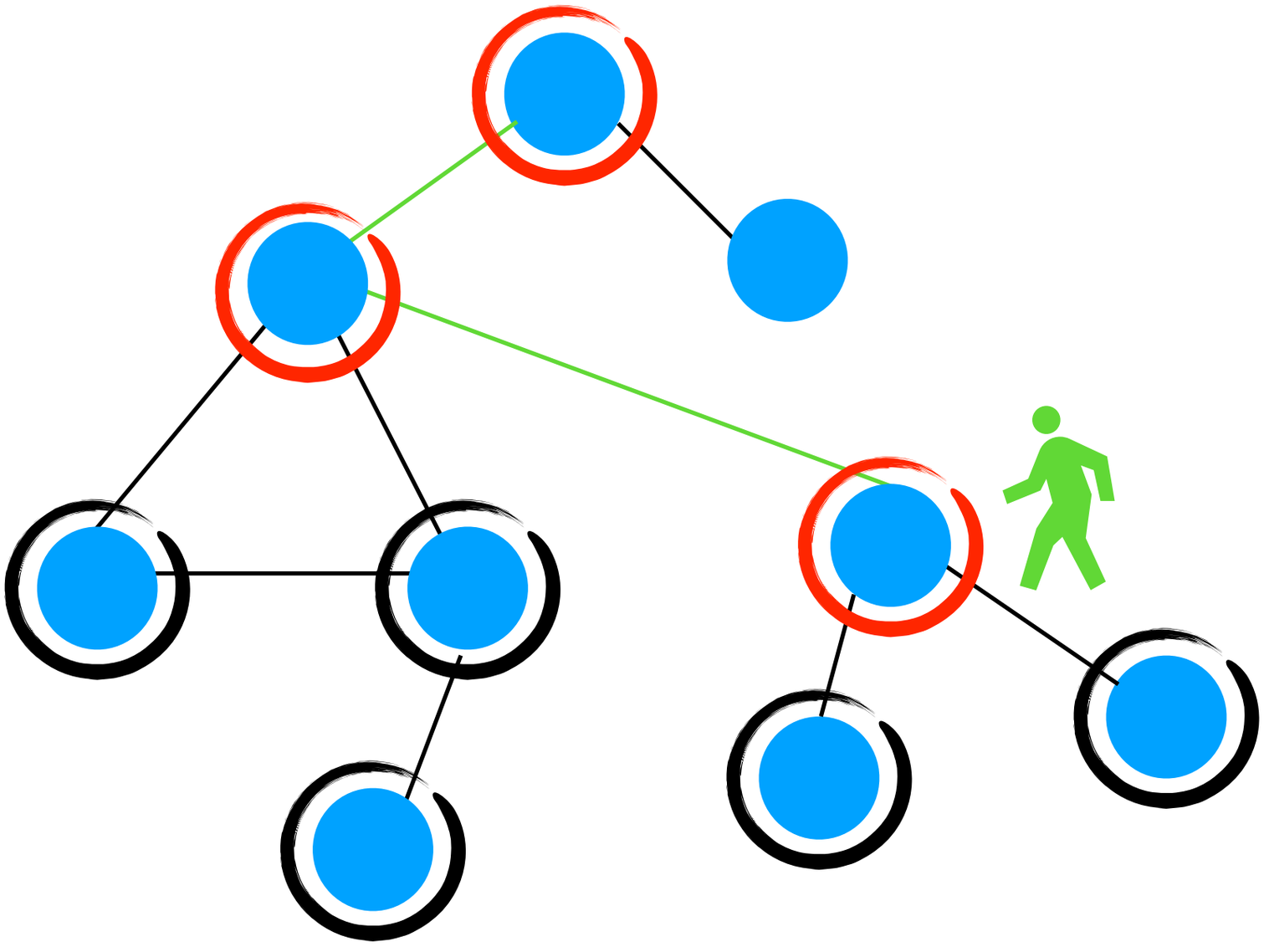
DFS



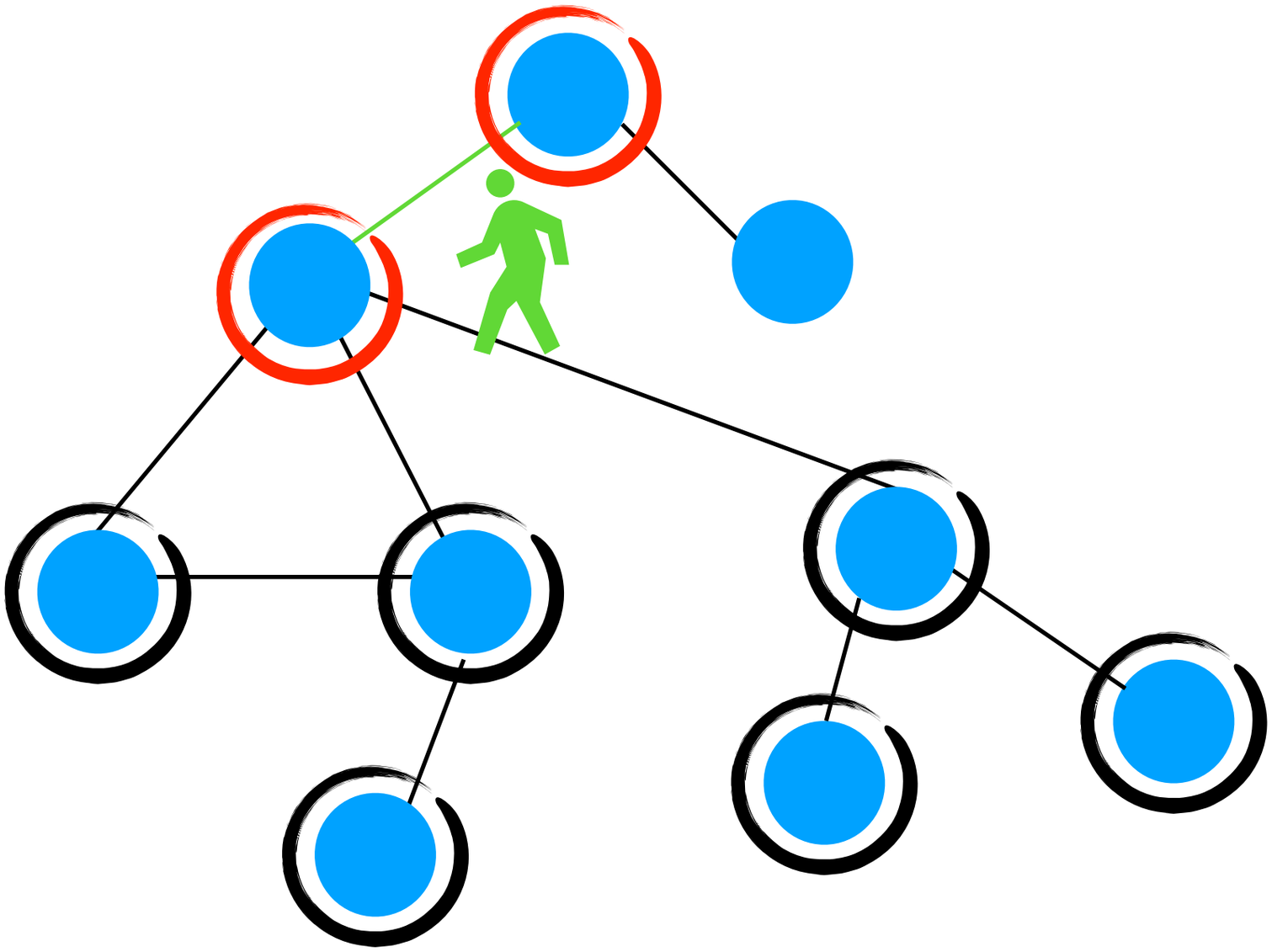
DFS



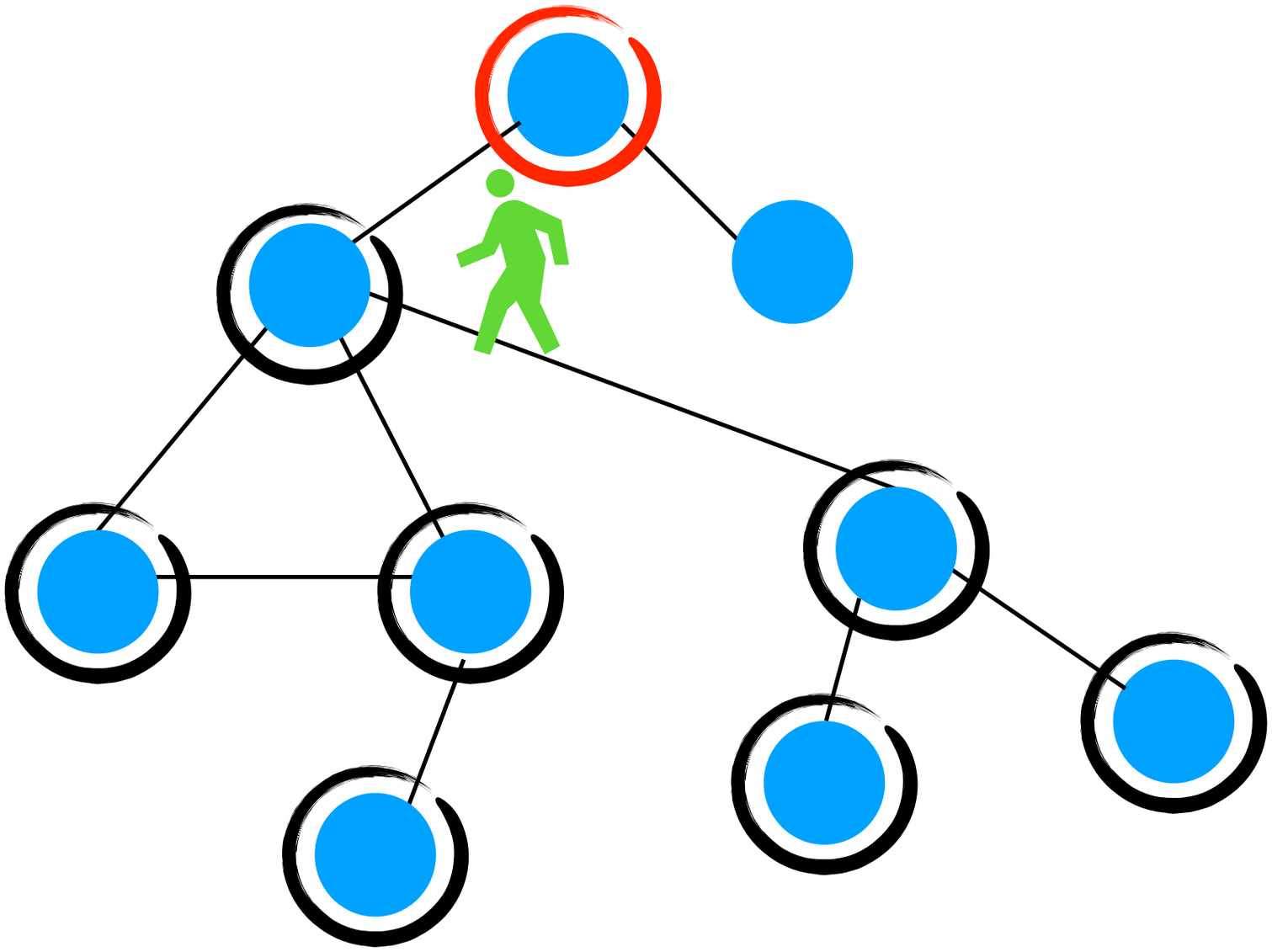
DFS



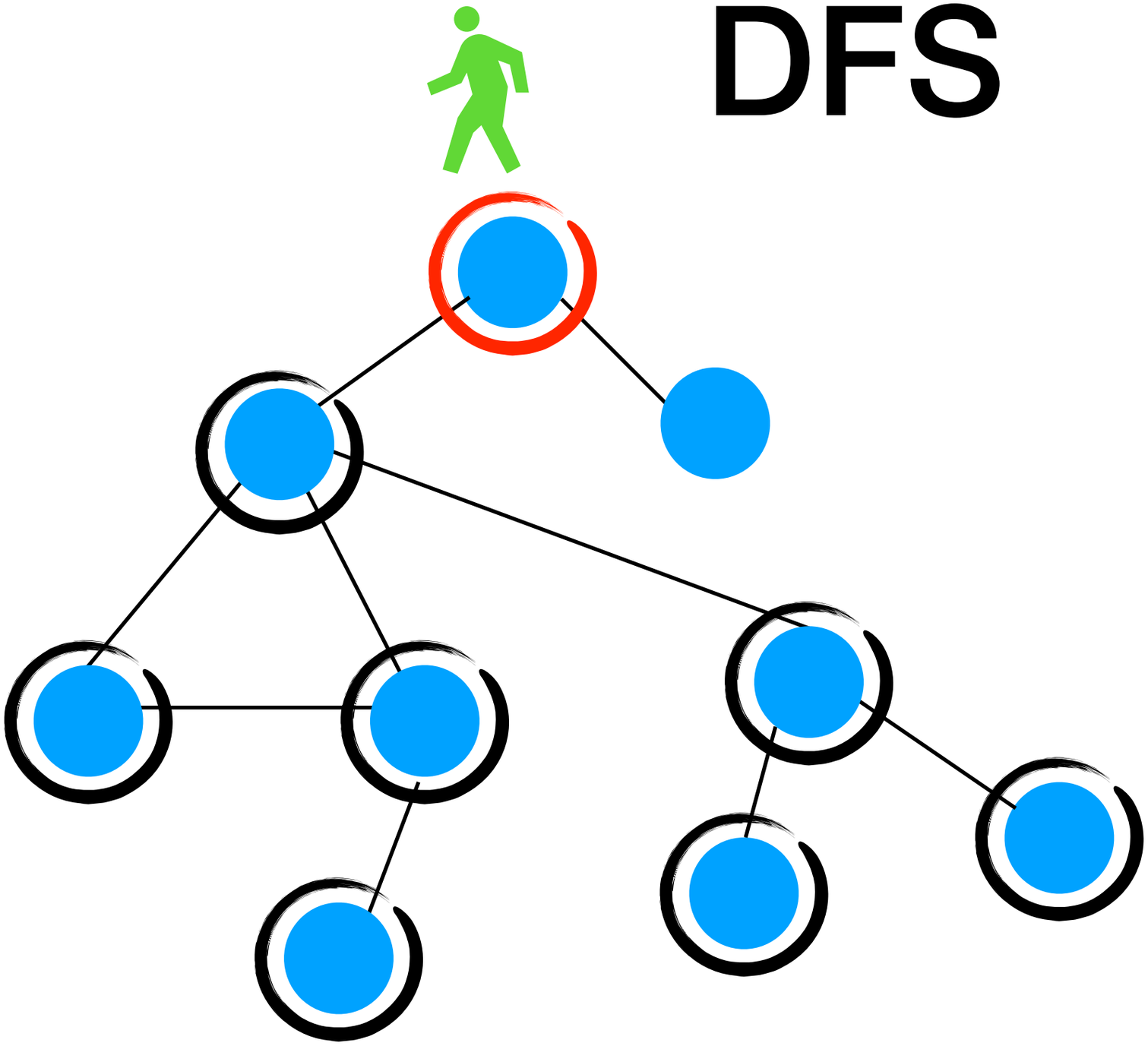
DFS



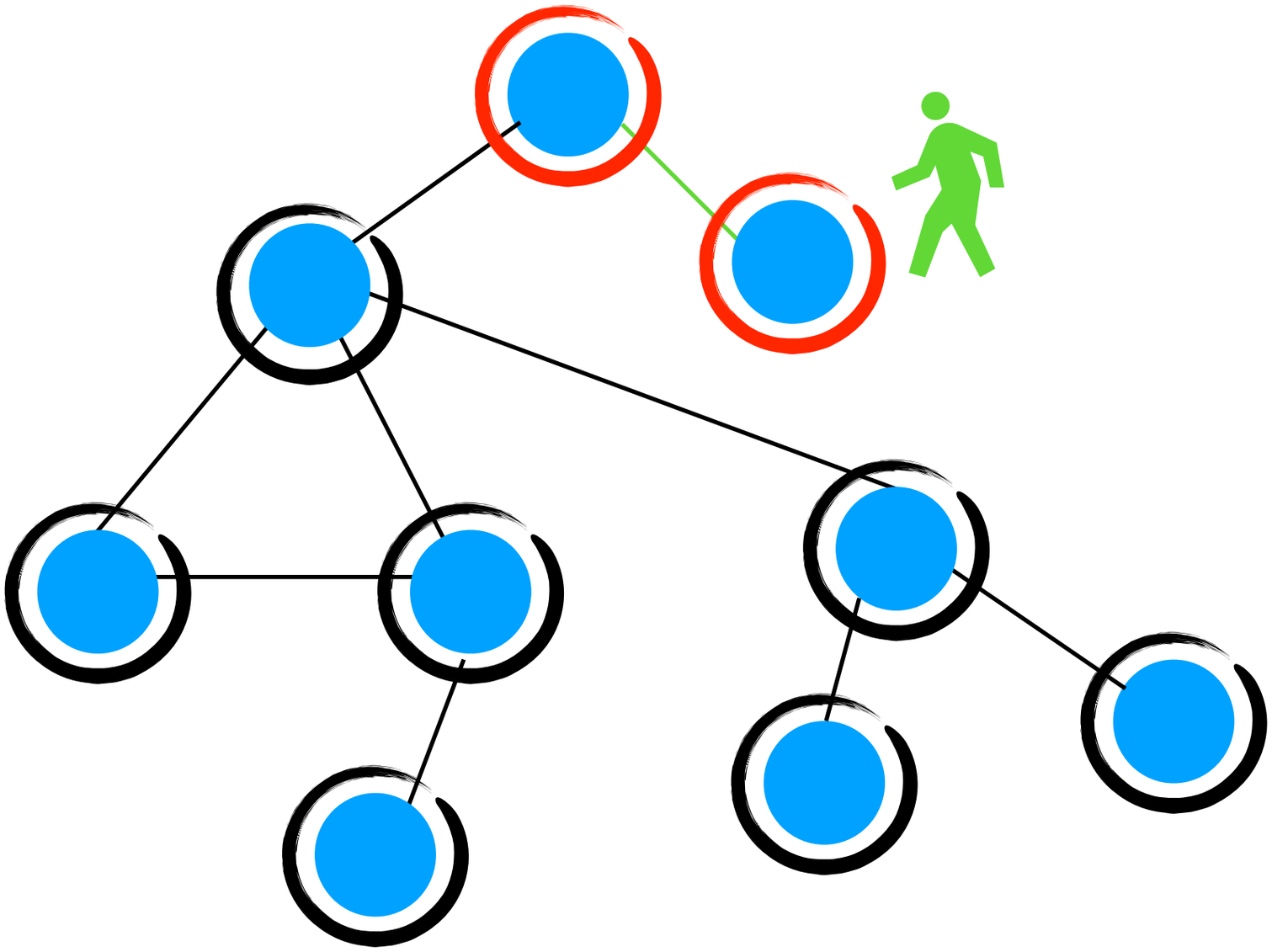
DFS



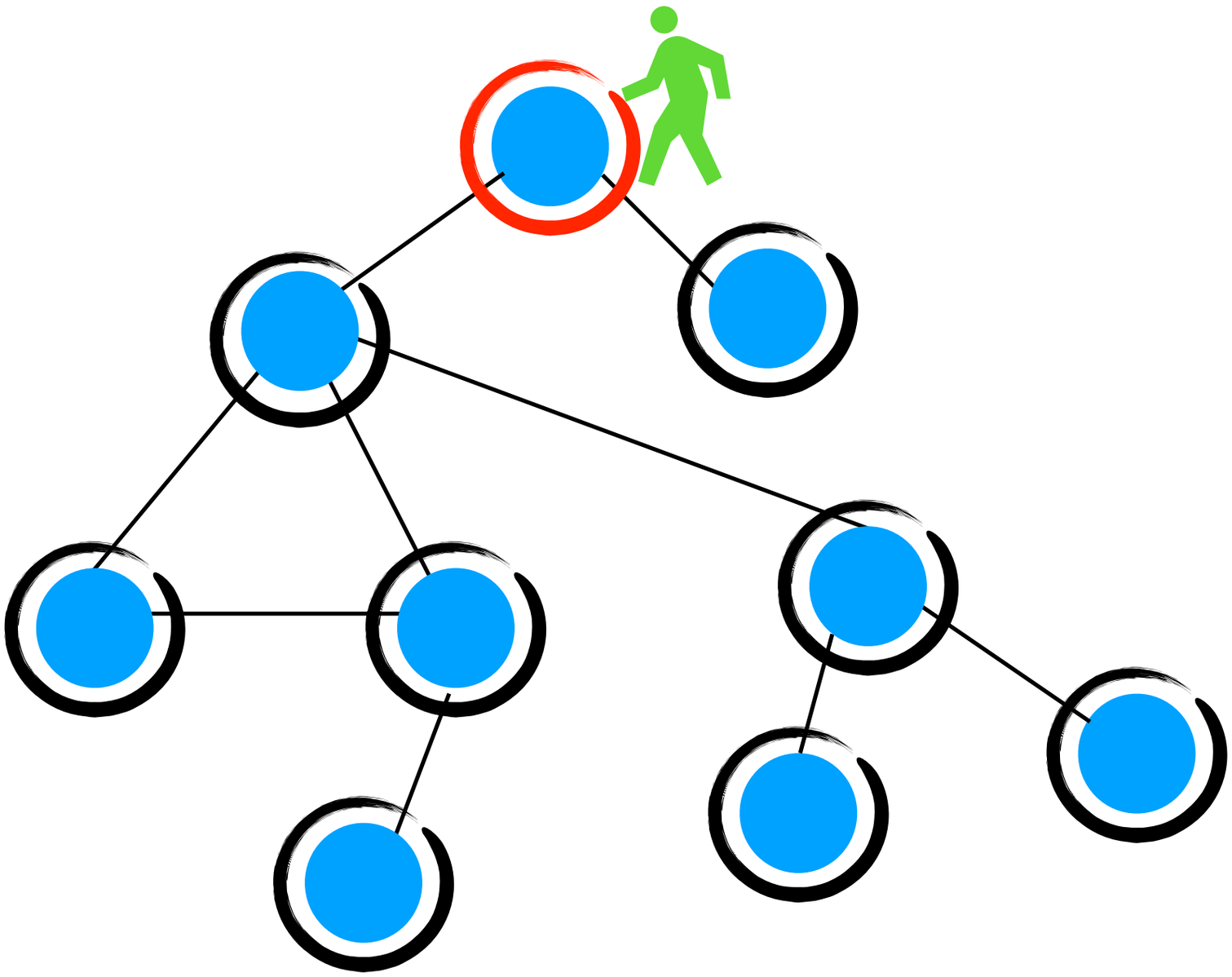
DFS



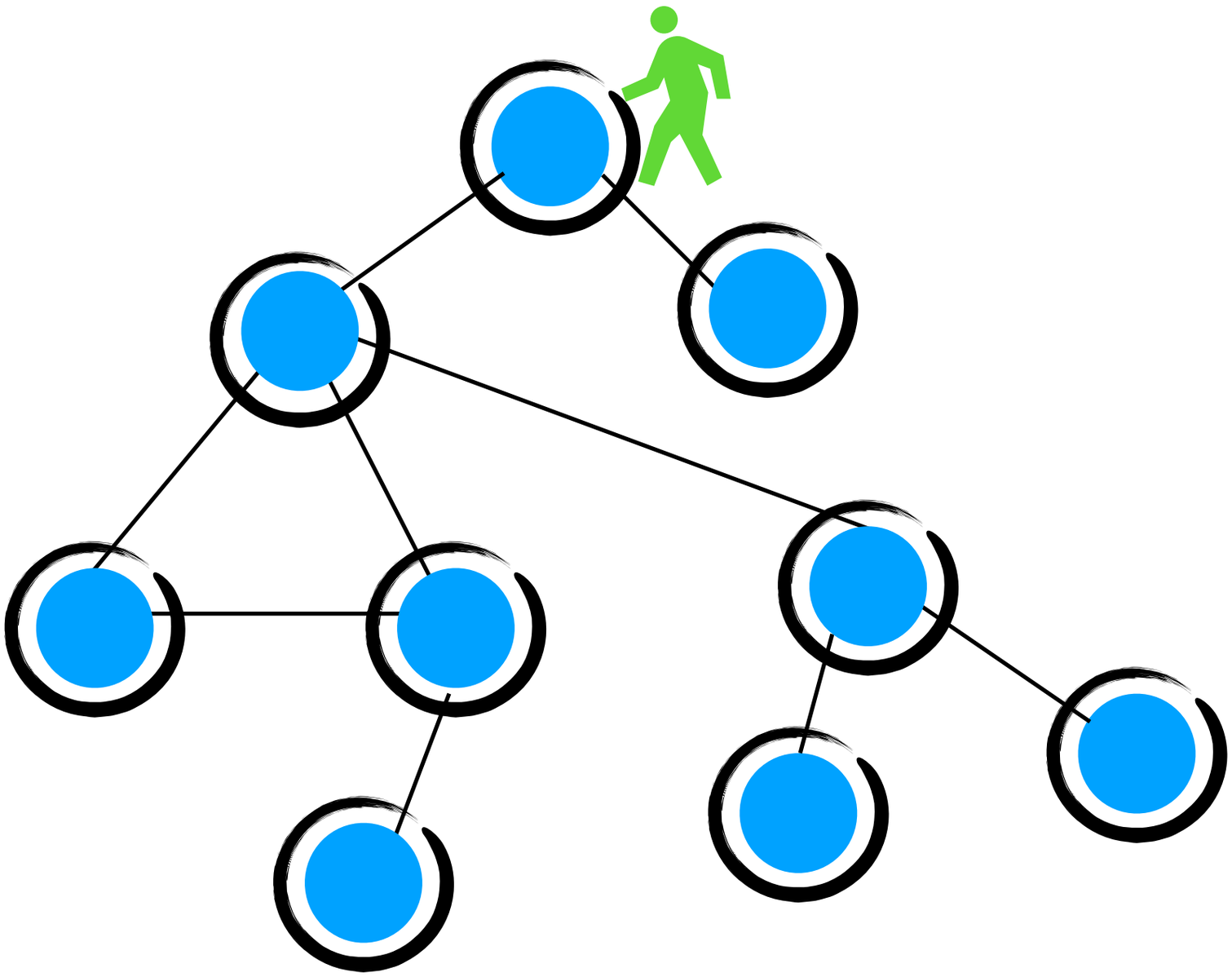
DFS



DFS



DFS



DFS: Java

```
  public static boolean[] used;
   int n = 6; //количество вершин
   x - это индекс вершины, в которой мы сейчас
   int[][] v - матрица смежности

public static int [] [] v;
   public static boolean[] used;

public static void dfs(int x) {
    used[x] = true;
    for (int i = 0; i < v[x].length; i++) {
        int to = i;
        if(!used[to] && v[x][to]==1){
            dfs(to);
        }
    }
}
```

DFS:C++

```
vector<bool> used;
vector< vector<int> > vec;
void dfs(int x){
    used[x] = true;
    for(int i = 0; i < vec[x].size(); i++) {
        int to = vec[x][i];
        if (!used[to])
            dfs(to);
    }
}
```

DFS: C

```c
void DFS(int i)
{
    int j;
    printf("\n%d",i);
    visited[i]=1;
    for(j=0;j<n;j++)
        if(!visited[j]&&G[i][j]==1)
            DFS(j);
}
```