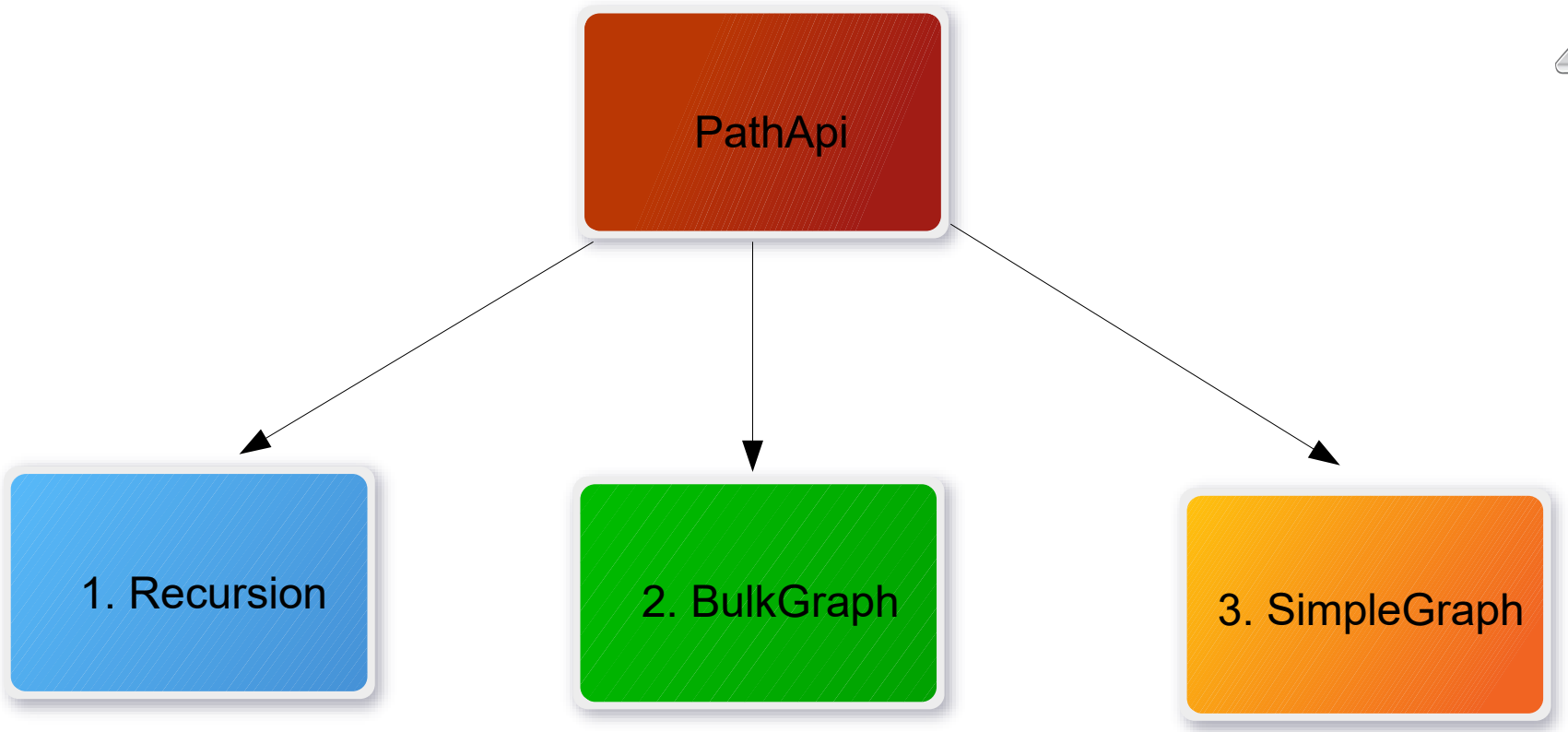
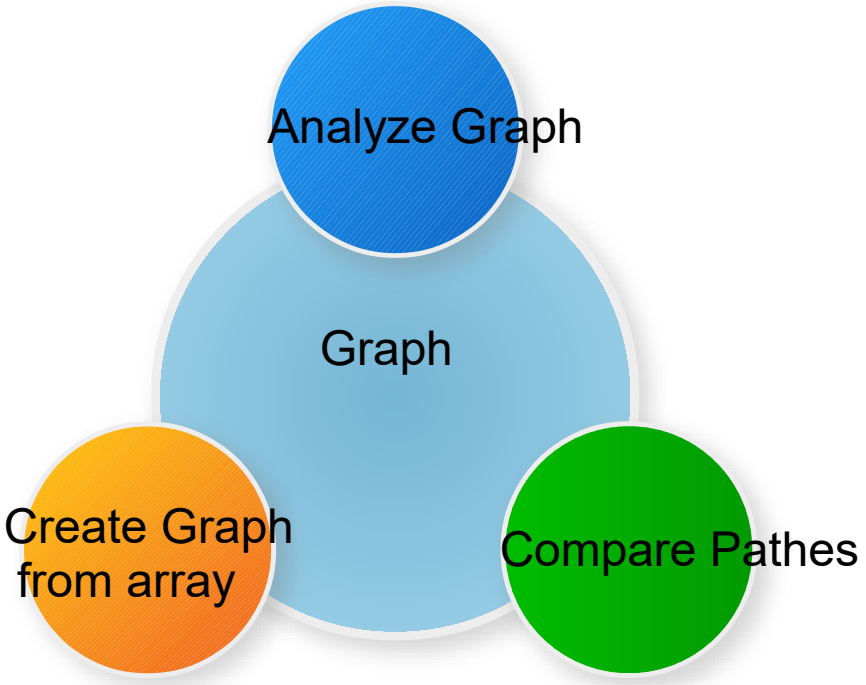



```
public int findMaxInPath(int[][] path);
```

Graph implementations



BulkGraph vs SimpleGraph

Bulk Graph	Simple Graph
Collects all Pathes and then looks for the most efficient Path	Compare Pathes while analyzing the Graph

Test cases

Tests implemented via JUnit

And check next cases:

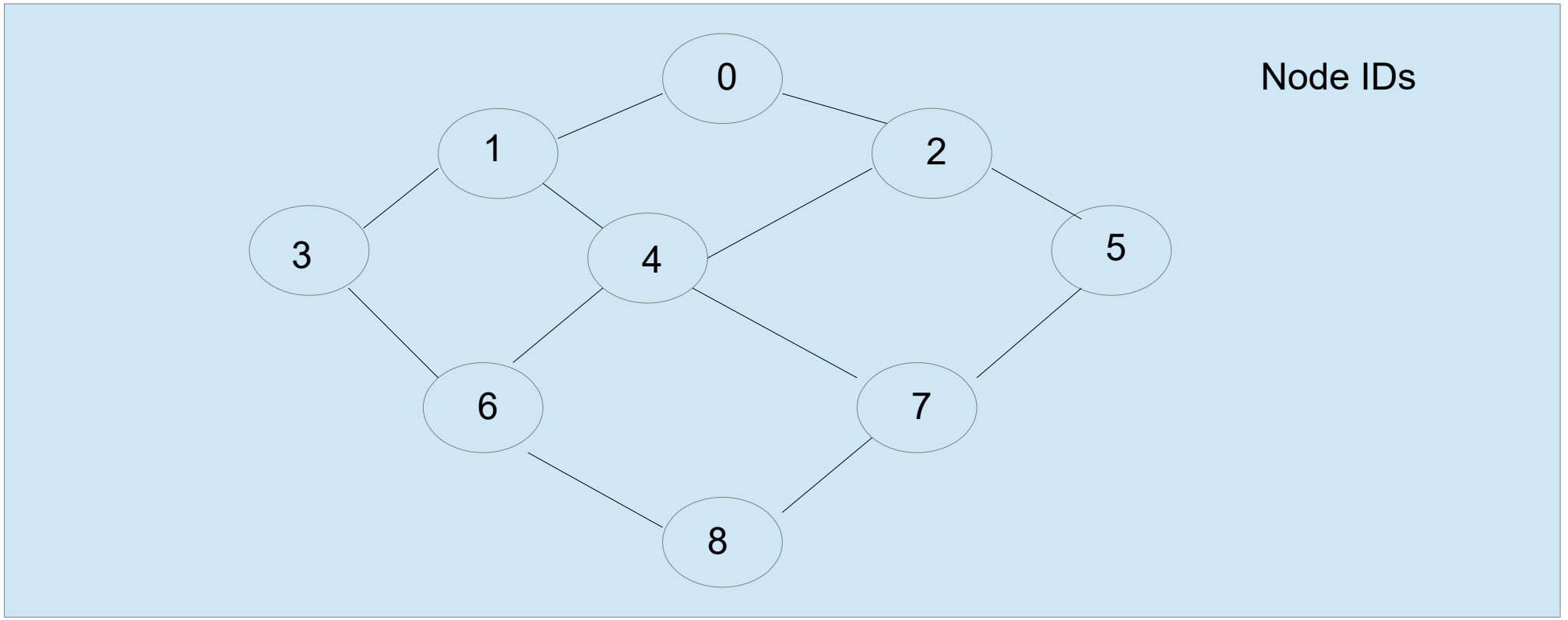
Case 1:

Node id in Graph

0	2	5
1	4	7
3	6	8

Array:

1	2	3
1	2	3
1	2	3



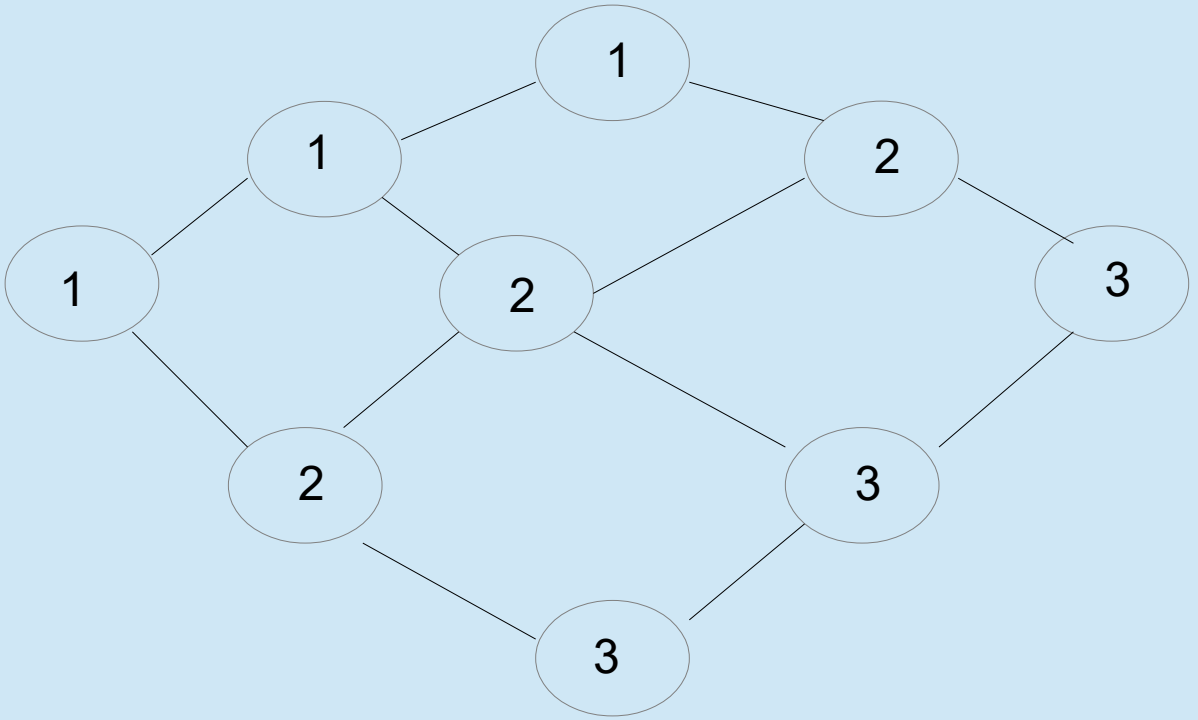
Case 1:

Node id in Graph

0	2	5
1	4	7
3	6	8

Array:

1	2	3
1	2	3
1	2	3



Array Values

Case 1. All available Pathes

1->2->3->3->3=12 is the best Path

1	1	1	2	3	8
1	1	2	2	3	9
1	1	2	2	3	9
1	1	2	3	3	10
1	2	2	3	3	11
1	2	3	3	3	12

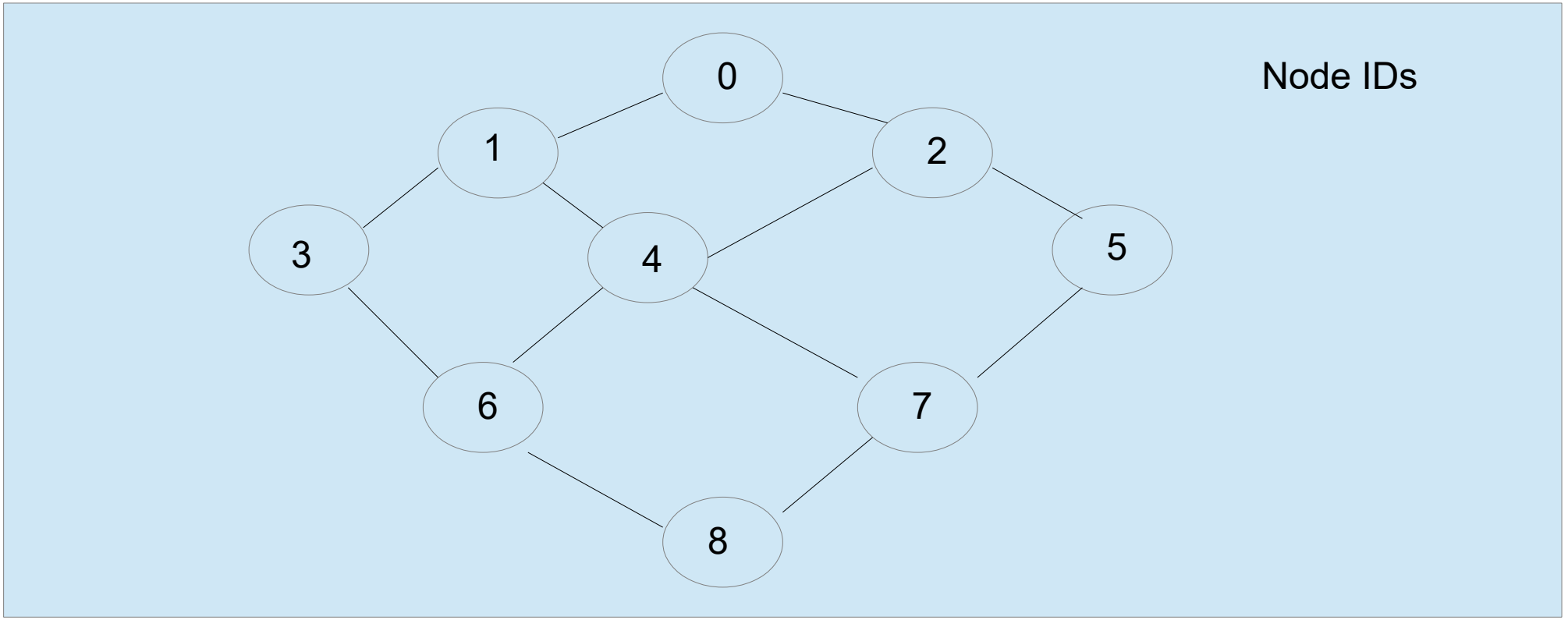
Case 2:

Node id in Graph

0	2	5
1	4	7
3	6	8

Array:

5	1	2
7	3	4
6	2	8



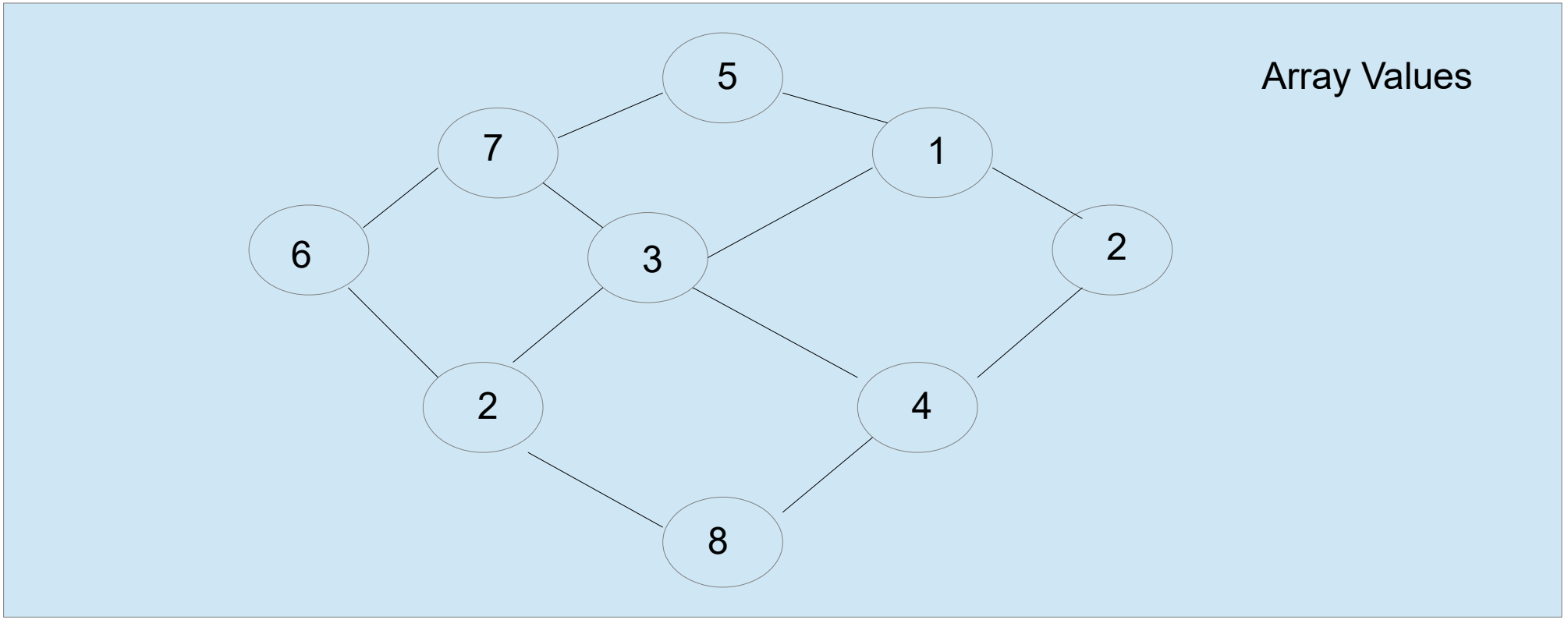
Case 2:

Node id in Graph

0	2	5
1	4	7
3	6	8

Array:

5	1	2
7	3	4
6	2	8



Case 2. All available Pathes

5->7->6->2->8=28 is the best Path

5	7	6	2	8	28
5	7	3	2	8	25
5	7	3	4	8	27
5	1	3	2	8	19
5	1	3	4	8	21
5	1	2	4	8	20

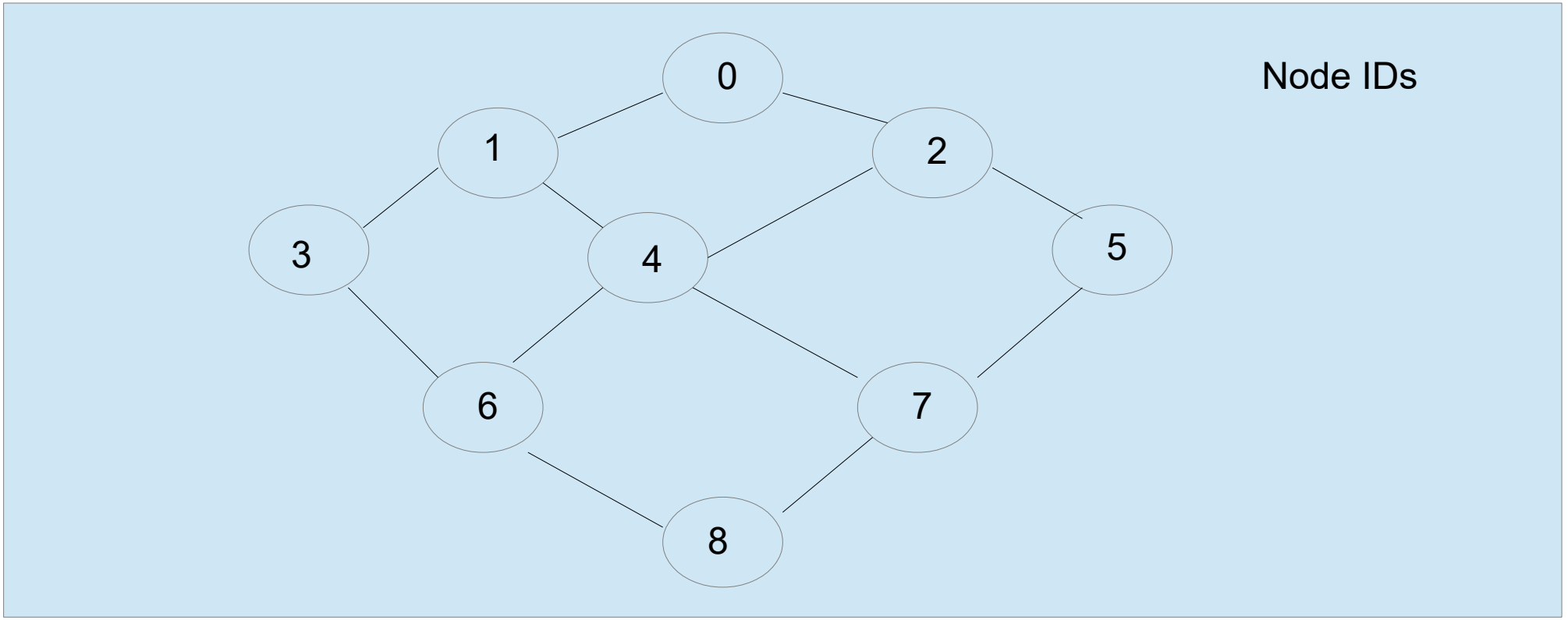
Case 3:

Node id in Graph

0	2	5
1	4	7
3	6	8

Array:

1	2	3
3	4	5
6	7	8



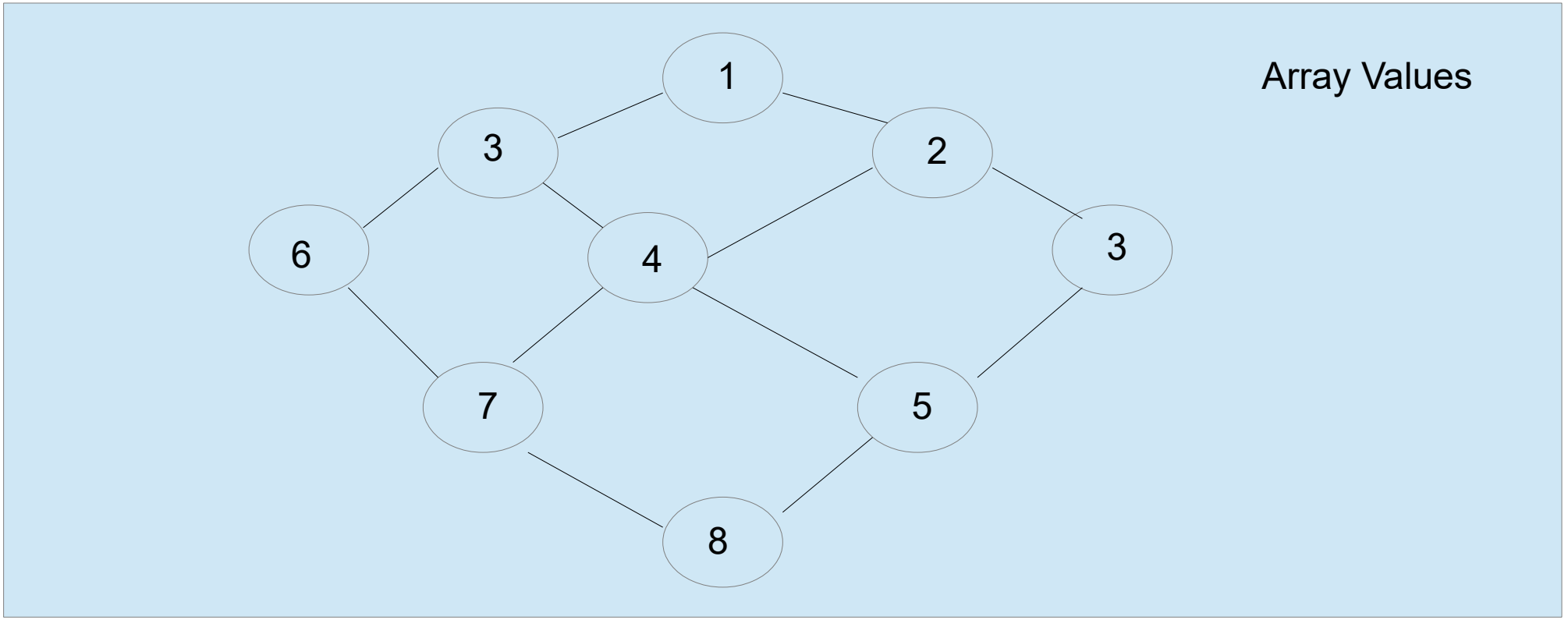
Case 3:

Node id in Graph

0	2	5
1	4	7
3	6	8

Array:

1	2	3
3	4	5
6	7	8



Case 2. All available Pathes

1->3->6->7->8=25 is the best Path

1	3	6	7	8	25
1	3	4	7	8	23
1	3	4	5	8	21
1	2	4	7	8	22
1	2	4	5	8	20
1	2	3	5	8	19