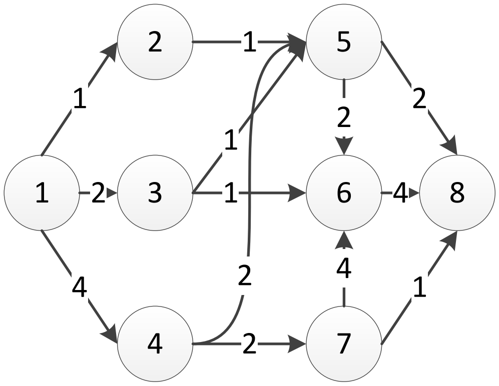
Example 1

Graph:



```
int graph[][] = {
        { 0, 1, 2, 4, 0, 0, 0, 0 },
        { 0, 0, 0, 0, 1, 0, 0, 0 },
        { 0, 0, 0, 0, 1, 1, 0, 0 },
        { 0, 0, 0, 0, 2, 0, 2, 0 },
        { 0, 0, 0, 0, 0, 2, 0, 2 },
        { 0, 0, 0, 0, 0, 0, 0, 4 },
        { 0, 0, 0, 0, 0, 4, 0, 1 },
        { 0, 0, 0, 0, 0, 0, 0, 0 } };
```

Result:

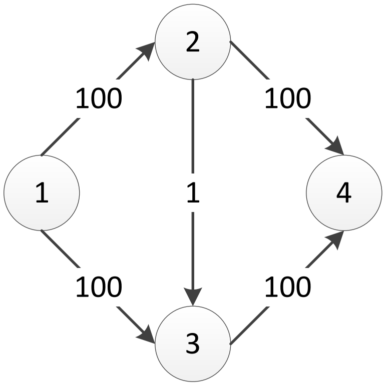
```
Method 1: use BFS on the residual graph:
5->8 2->5 1->2 flow: 1
5->8 3->5 1->3 flow: 1
6->8 3->6 1->3 flow: 1
7->8 4->7 1->4 flow: 1
6->8 5->6 4->5 1->4 flow: 2
6->8 7->6 4->7 1->4 flow: 1
The maximum flow: 7

Method 2: use DFS on the residual graph:
7->8 4->7 1->4 flow: 1
6->8 7->6 4->7 1->4 flow: 1
5->8 4->5 1->4 flow: 2
6->8 3->6 1->3 flow: 1
6->8 5->6 3->5 1->3 flow: 1
6->8 5->6 2->5 1->2 flow: 1
The maximum flow: 7

Method 3: use DFS on the original graph:
6->8 flow: 1 5->6 flow: 1 2->5 flow: 1 1->2 flow: 1
6->8 flow: 1 5->6 flow: 1 3->5 flow: 1 1->3 flow: 1
5->8 flow: 1 6->5 flow: 1 3->6 flow: 1 1->3 flow: 1
6->8 flow: 1 5->6 flow: 1 4->5 flow: 1 1->4 flow: 1
5->8 flow: 1 4->5 flow: 1 1->4 flow: 1
6->8 flow: 1 7->6 flow: 1 4->7 flow: 1 1->4 flow: 1
7->8 flow: 1 4->7 flow: 1 1->4 flow: 1
The maximum flow: 7
```

Example 2

Graph:



```
int graph[][] = {
        { 0, 1000, 1000, 0 },
        { 0, 0, 1, 1000 },
        { 0, 0, 0, 1000 },
        { 0, 0, 0, 0 } };
```

Result:

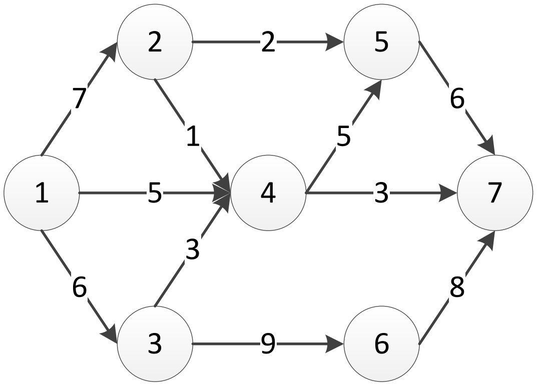
```
Method 1: use BFS on the residual graph:
2->4 1->2 flow: 1000
3->4 1->3 flow: 1000
The maximum flow: 2000

Method 2: use DFS on the residual graph:
3->4 1->3 flow: 1000
2->4 1->2 flow: 1000
The maximum flow: 2000

Method 3: use DFS on the original graph:
3->4 flow: 1 2->3 flow: 1 1->2 flow: 1
2->4 flow: 999 1->2 flow: 999
2->4 flow: 1 3->2 flow: 1 1->3 flow: 1
3->4 flow: 999 1->3 flow: 999
The maximum flow: 2000
```

Example 3

Graph:



```
int graph[][] = {
          { 0, 7, 6, 5, 0, 0, 0 },
          { 0, 0, 0, 1, 2, 0, 0 },
          { 0, 0, 0, 3, 0, 9, 0 },
          { 0, 0, 0, 0, 5, 0, 3 },
          { 0, 0, 0, 0, 0, 0, 6 },
          { 0, 0, 0, 0, 0, 0, 8 },
          { 0, 0, 0, 0, 0, 0, 0 } };
```

Result:

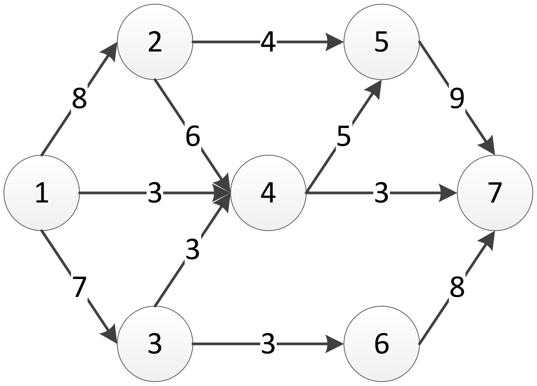
```
Method 1: use BFS on the residual graph:
4->7 1->4 flow: 3
5->7 2->5 1->2 flow: 2
6->7 3->6 1->3 flow: 6
5->7 4->5 1->4 flow: 2
5->7 4->5 2->4 1->2 flow: 1
The maximum flow: 14

Method 2: use DFS on the residual graph:
4->7 1->4 flow: 3
5->7 4->5 1->4 flow: 2
6->7 3->6 1->3 flow: 6
5->7 2->5 1->2 flow: 2
5->7 4->5 2->4 1->2 flow: 1
The maximum flow: 14

Method 3: use DFS on the original graph:
5->7 flow: 1 4->5 flow: 1 2->4 flow: 1 1->2 flow: 1
4->7 flow: 1 5->4 flow: 1 2->5 flow: 1 1->2 flow: 1
5->7 flow: 1 2->5 flow: 1 1->2 flow: 1
5->7 flow: 3 4->5 flow: 3 3->4 flow: 3 1->3 flow: 3
6->7 flow: 3 3->6 flow: 3 1->3 flow: 3
6->7 flow: 3 3->6 flow: 3 4->3 flow: 3 1->4 flow: 3
5->7 flow: 1 4->5 flow: 1 1->4 flow: 1
4->7 flow: 1 1->4 flow: 1
The maximum flow: 14
```

Example 4

Graph:



```
int graph[][] = {
          { 0, 8, 7, 3, 0, 0, 0 },
          { 0, 0, 0, 6, 4, 0, 0 },
          { 0, 0, 0, 3, 0, 3, 0 },
          { 0, 0, 0, 0, 5, 0, 3 },
          { 0, 0, 0, 0, 0, 0, 9 },
          { 0, 0, 0, 0, 0, 0, 8 },
          { 0, 0, 0, 0, 0, 0, 0 } };
```

Result:

```
Method 1: use BFS on the residual graph:
4->7 1->4 flow: 3
5->7 2->5 1->2 flow: 4
6->7 3->6 1->3 flow: 3
5->7 4->5 2->4 1->2 flow: 4
5->7 4->5 3->4 1->3 flow: 1
The maximum flow: 15

Method 2: use DFS on the residual graph:
4->7 1->4 flow: 3
6->7 3->6 1->3 flow: 3
5->7 4->5 3->4 1->3 flow: 3
5->7 2->5 1->2 flow: 4
5->7 4->5 2->4 1->2 flow: 2
The maximum flow: 15

Method 3: use DFS on the original graph:
5->7 flow: 5 4->5 flow: 5 2->4 flow: 5 1->2 flow: 5
4->7 flow: 1 2->4 flow: 1 1->2 flow: 1
4->7 flow: 2 5->4 flow: 2 2->5 flow: 2 1->2 flow: 2
5->7 flow: 2 2->5 flow: 2 4->2 flow: 2 3->4 flow: 2 1->3 flow: 2
5->7 flow: 1 4->5 flow: 1 3->4 flow: 1 1->3 flow: 1
6->7 flow: 3 3->6 flow: 3 1->3 flow: 3
5->7 flow: 1 4->5 flow: 1 1->4 flow: 1
The maximum flow: 15
```