Hint1:

Input:

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

Output:

```
6
'\n'
```

##

In this assignment, our output has a newline '\n'.
That means when we output the result, we need to output like:
cout<<"C"<<endl;

If I show '\b' in later assignments,
That means blank space:
cout<<" "; or cin>>" ";

Please using the example_data to check your answer is right or not before you upload the code.

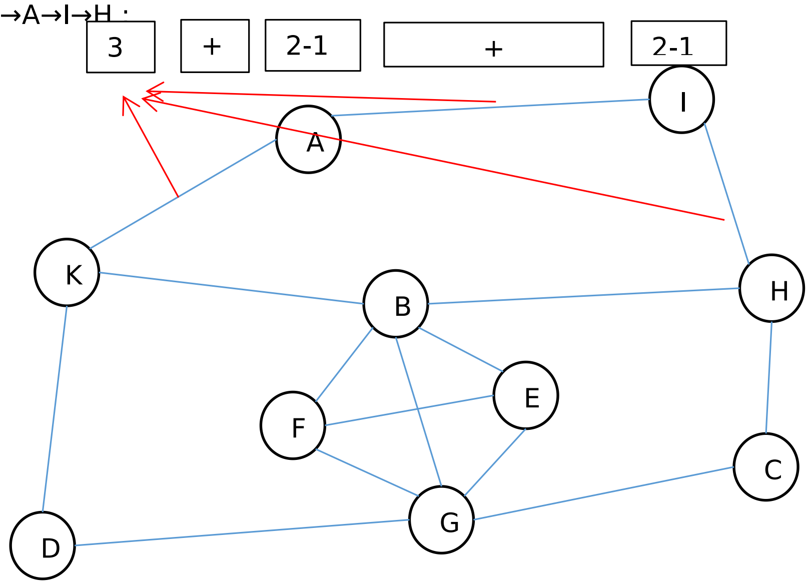
'graph' folder is the inputs' corresponding graph.

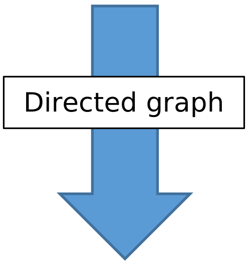
##

Hint2:

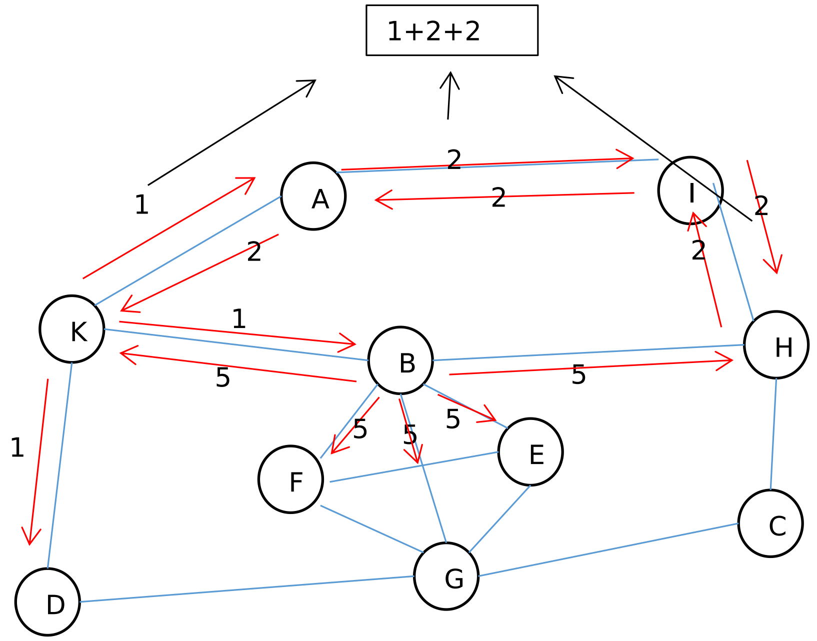
In my opinion, it will be faster to convert undirected graph into   and use **Dijkstra** algorithm to solve the problem. It's just my opinion. You don't need to follow my advice.

Like K→A→I→H :





1. K is source node, so all outgoing edge's weight change to 1;
2. A,I is midian point, all outgoing edge's weight is the the number of undirected edges of this node.
3. Using **Dijkstra** to find the shortest path from K to H.



1+2+2
1
2
2
2
2
2
1
5
5
5
5
5
1
A
I
K
B
H
F
E
C
G
D