Uva 1234 (LA4110) Racing

Times: 3 seconds

Problem Descriptions (1/5)

- Singapore will host a Formula One race in 2008. The race will be held on a 5.067km long street circuit, consisting of 14 left hand turns and 10 right hand turns.
- In the run up to the F1 race, the number of illegal night street racing activities have been on the rise. Such races consists of several rounds around a designated street circuit.

Problem Descriptions (2/5)

- **The authorities would like to deploy a new vehicle monitoring system in order to catch these illegal Saint Andrew's Road, part of the Formula One circuit racers.**
- **The system consists of a (Kenny Pek, Piccom) number of cameras mounted along various roads. For the system to be effective, there should be <u>at least one camera along each of the possible circuits</u>.**

Problem Descriptions (3/5)

- The Singapore road system can be represented as a series of junctions and connecting bidirectional roads (see Figure 5).
- A possible racing circuit consists of a start junction followed by a path consisting of three or more roads that eventually leads back to the start junction.
- Each road in a racing circuit can be traversed only in one direction, and only once.

Problem Descriptions (4/5)

- **Your task is to write a program that <u>computes the optimal placement of the vehicle-monitoring cameras</u>.**
- **You will be provided with a description of a connected road network to be monitored in terms of the roads and junctions.**
  - **The junctions are <u>identified by the bigger numbers</u> in Figure 5.**
  - **A <u>camera can be deployed on the roads</u> (and not the junctions).**

Problem Descriptions (5/5)

- **The <u>cost of deploying a camera depends on the road on which it is placed</u>.**
- **The smaller numbers by the roads in Figure 5 indicate the cost of deploying a camera on that road.**
- **Your job is to select <u>a set of roads that minimizes the total cost</u> of deployment <u>while ensuring that there is at least one camera along every possible racing circuit</u> (i.e. loop in the road network).**

Input

- The input consists of a line containing the number c of datasets, followed by c datasets, followed by a line containing the number ‘0’.
- The first line of each dataset contains two positive integers, n and m, separated by a blank, which represent the number of junctions and number of roads, respectively.

Input

- You may assume that
  - $0 < n < 10000$ and $0 < m < 100000$
- For simplicity, we label each of the n junctions from 1 to n. The following m lines of each dataset each describes one road. Each line consists of three positive integers which are the labels of two different junctions and the cost of deploying a camera on this road.
- The cost of deploying a camera is between 1 and 1000.

Output

- The output consists of one line for each dataset.
- The c-th line contains one single number, representing the minimal cost of setting up the vehicle monitoring system such that there is at least one camera along every possible circuit.

Note:

1) The sample data set depicts the situation shown in Figure 5.
2) The two cameras show where cameras might be placed in order to monitor each circuit at minimal cost.
3) Since each of the cameras have a cost of 3, the total minimal cost is 6.

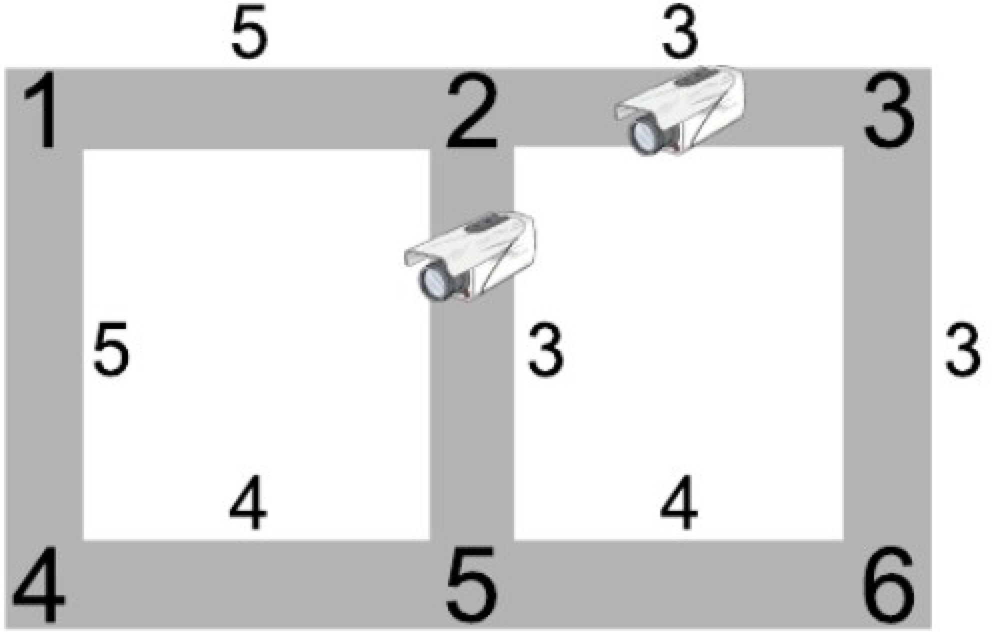


Figure 5: Illustration of sample input and one possible optimal placement of cameras

I/O Example

Input

Number of test case

1

6 7

n,m

Number of junctions and roads

1 2 5

2 3 3

1 4 5

4 5 4

5 6 4

6 3 3

5 2 3

m roads

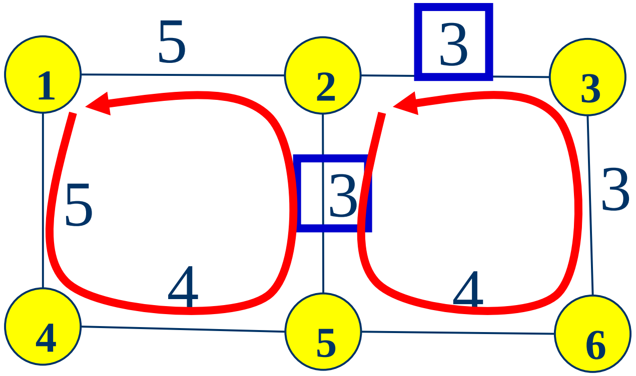
node-node-cost

0

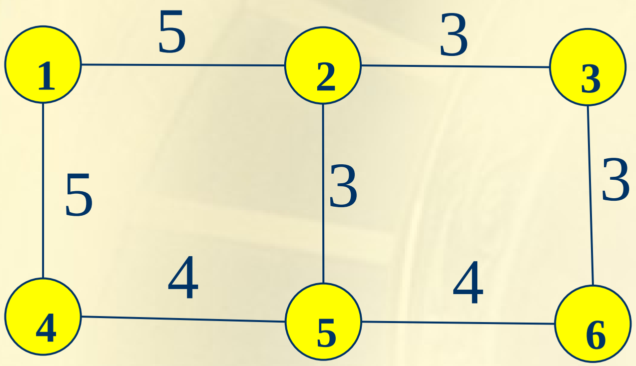
end of test case

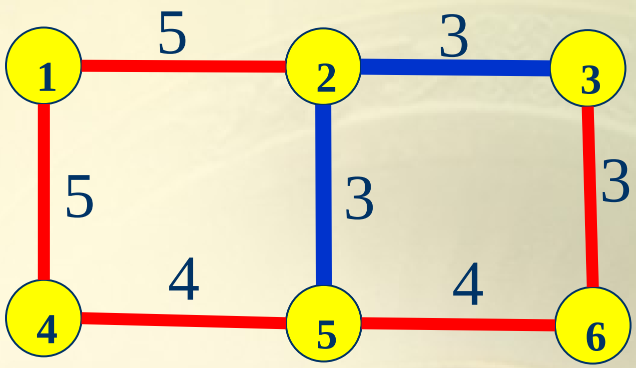
Output

6



Maximum Spanning Tree



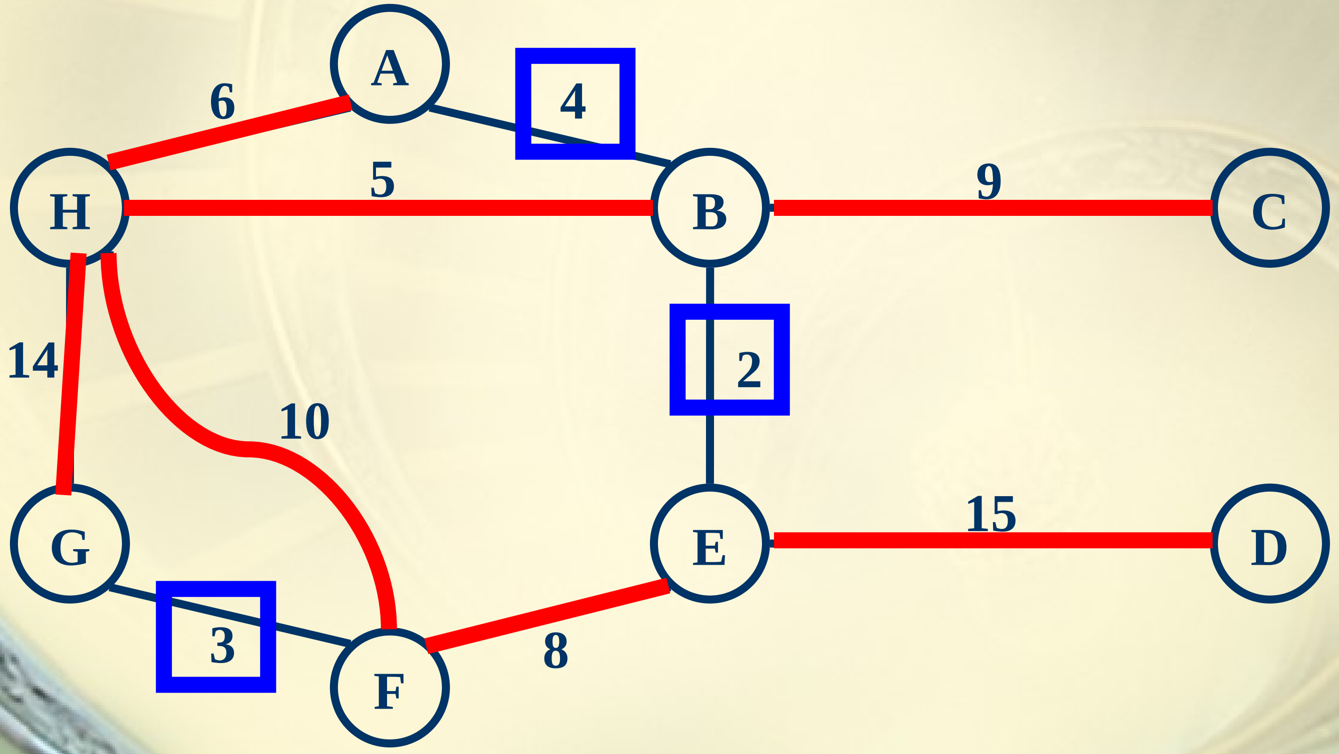


Answer: 3+3=6

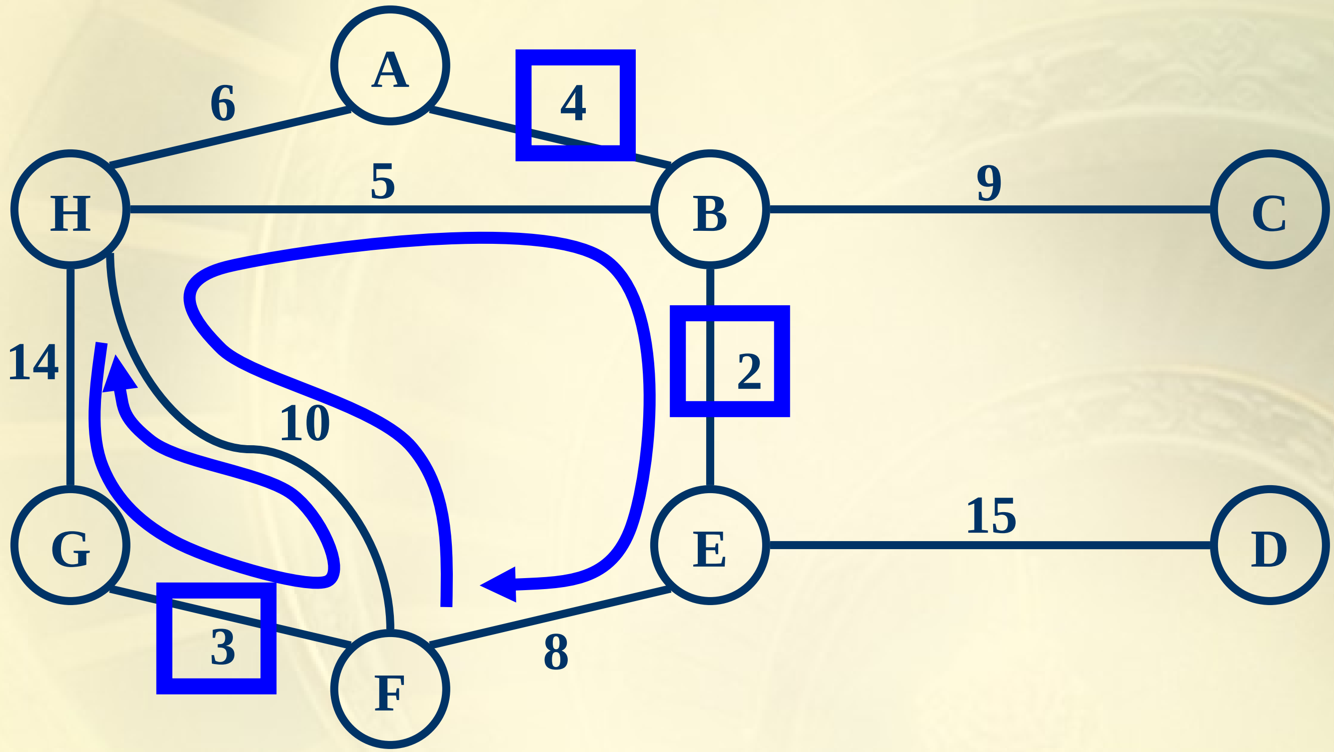
Kruskal's Algorithm

- Answer: <u>Maximum Cost=67</u>

Sort Edges: 2 | 3 | 4 | 5 | 6 | 8 | 9 | 10 | 14 | 15 |



Set Cameras



Kruskal’s Algorithm

Kruskal’s Algorithm(G(V,E))

```
{ T←
  While |T|<|V|-1 and E≠
  { ExtraxtMax (v,w) from E
    If(v,w)doesn’t cause cycle in T
        then add (v,w) to T
        else discard (v,w)
  }
}
```