

- Overview:** Find the minimum weight vertex cover of a graph with no cycles.
- Description:** Despite Loki's shenanigans in the previous problem, all is not lost! Agent Maria Hill has developed a clever counter that treats New York City like an undirected graph with vertices and edges (this graph has no cycles). Vertices have weights and represent the importance of a particular location (for example, the local bagel shop has even higher weight than even the Statue of Liberty!). Edges represent important roads and connections between vertices.
- Really, if the roads and connections are preserved, everything will be okay and Loki will be unable to land black holes everywhere (because all he sees are roads). However, Maria Hill's mysterious, all-powerful counter must be installed at vertices (locations in New York), and the monetary cost of installing the counter is the vertex weight. Counters on a vertex protect all edges emanating from the vertex. You've been asked to help figure out which vertices must have counters installed so that Maria Hill can protect New York at the cheapest cost possible. Are you up for the challenge?
- Filename:** adv93.{java, cpp, c, cc, py}
- Input:** The first line contains an integer T , the number of test cases in the input file. After this comes each of the test cases.
- The first line of a test case contains two integers, N and E . N is the number of vertices in the graph and E is the number of edges in the graph.
- The next line contains N integers, the weights of the N vertices, ordered by number from 1 to N .
- Each of the next E lines contains two integers, A and B , implying that vertices A and B are connected by an edge.
- Output:** For each test case, output a single integer with the minimum sum of weights that covers every edge in the graph.
- Assumptions:** The number of vertices in the graph is no more than 5000.
 $1 \leq A, B \leq N$
 The graph contains no cycles.
 The weight of a vertex is a positive integer no more than 1000.
- Sample Input #1:**
- ```

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

```

Sample 4

Output #1: 1