|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| http://ace.delos.com/usaco/cowhead2.gif | |  | | --- | | Contest: OPEN11 **GOLD** Division | |  | |  | |  | |

**ANALYSIS MODE  
Submit solutions for your own enjoyment.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**GOLD PROBLEMS**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Three problems numbered 1 through 3**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Problem 1: Mowing the Lawn [Neal Wu, 2008]**

**After winning the annual town competition for best lawn a year ago,**

**Farmer John has grown lazy; he has not mowed the lawn since then**

**and thus his lawn has become unruly. However, the competition is**

**once again coming soon, and FJ would like to get his lawn into**

**tiptop shape so that he can claim the title.**

**Unfortunately, FJ has realized that his lawn is so unkempt that he**

**will need to get some of his N (1 <= N <= 100,000) cows, who are**

**lined up in a row and conveniently numbered 1..N, to help him. Some**

**cows are more efficient than others at mowing the lawn; cow i has**

**efficiency E\_i (0 <= E\_i <= 1,000,000,000).**

**FJ has noticed that cows near each other in line often know each**

**other well; he has also discovered that if he chooses more than K**

**(1 <= K <= N) consecutive (adjacent) cows to help him, they will**

**ignore the lawn and start a party instead. Thus, FJ needs you to**

**assist him: determine the largest total cow efficiency FJ can obtain**

**without choosing more than K consecutive cows.**

**PROBLEM NAME: mowlawn**

**INPUT FORMAT:**

**\* Line 1: Two space-separated integers: N and K**

**\* Lines 2..N+1: Line i+1 contains the single integer: E\_i**

**SAMPLE INPUT (file mowlawn.in):**

**5 2**

**1**

**2**

**3**

**4**

**5**

**INPUT DETAILS:**

**FJ has 5 cows whose efficiencies are 1, 2, 3, 4, and 5, in that**

**order. He wants to choose some of the cows such that their total**

**efficiency is maximized, but he cannot choose more than 2 consecutive**

**cows.**

**OUTPUT FORMAT:**

**\* Line 1: A single integer that is the best total efficiency FJ can**

**obtain.**

**SAMPLE OUTPUT (file mowlawn.out):**

**12**

**OUTPUT DETAILS:**

**FJ chooses all cows but the third. The total efficiency of the cows is thus**

**1 + 2 + 4 + 5 = 12.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Problem 2: Odd degrees [Traditional, 2011]**

**The cows are being invaded! Their republic comprises N (1 <= N <=**

**50,000) towns that are connected by M (1 <= M <= 100,000) undirected**

**paths between two towns A\_i and B\_i (1 <= A\_i <= N; 1 <= B\_i <= N;**

**A\_i != B\_i; no duplicate paths will occur). However the republic**

**is not necessarily connected--there may be pairs of towns that are**

**unable to reach each other through the paths.**

**The cows know their invaders plan to conduct an inventory of every**

**path within their republic, so they are willing to shut down various**

**paths to make it as difficult as possible for their invaders to do**

**so.**

**Please help the cows find a way to shut down a subset of the paths**

**such that each town has an odd number of remaining paths connected**

**to it, or determine if no such subset exists.**

**For example, consider the following cow republic:**

**1---2**

**\ /**

**3---4**

**If we keep the paths 1-3, 2-3, and 3-4, and remove the path 1-2, then towns**

**1, 2, and 4 will be an endpoint of exactly one path, whereas town 3 will be**

**an endpoint of three paths:**

**1 2**

**\ /**

**3---4**

**PROBLEM NAME: oddd**

**INPUT FORMAT:**

**\* Line 1: Two space-separated integers: N and M**

**\* Lines 2..M+1: Line i+1 contains two space-separated integers: A\_i**

**and B\_i**

**SAMPLE INPUT (file oddd.in):**

**4 4**

**1 2**

**2 3**

**3 1**

**3 4**

**OUTPUT FORMAT:**

**\* Line 1: A single integer that is the number of paths to keep. If no**

**subset exists output only a single line with the integer -1.**

**\* Lines 2..K+1: Each line contains an index of an path to keep, in the**

**range 1..M. These indices must be pairwise distinct.**

**SAMPLE OUTPUT (file oddd.out):**

**3**

**2**

**3**

**4**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Problem 3: Soldering [Michael Cohen, 2011]**

**The cows are playing with wires! They have learned a technique**

**called soldering, in which they connect two pieces of wire together**

**by attaching the endpoint of one wire to a location along the length**

**of the other. (Soldering endpoint to endpoint is not allowed.) There**

**can be multiple solder junctions at the same point.**

**The cows have a plan for an Amazing Structure they would like to**

**build. It is in the form of a graph with N (1 <= N <= 50,000) nodes**

**and N-1 edges of unit length so that each pair of nodes is connected.**

**Each edge is described by a pair of integers, A and B (1 <= A <=**

**N; 1 <= B <= N; A != B).**

**The cows are able to buy wire from a local store; however longer**

**wire is more expensive. In particular the cows can buy a wire of**

**length L with cost L\*L, but they cannot cut wires or join wires**

**together.**

**Given the plan, the cows would like solder wires together to build**

**their Amazing Structure. Please help them find the minimum possible**

**cost!**

**Test data worth at least 50% of the points will have N <= 2,000.**

**Partial feedback will be provided on your first 50 submissions to this problem.**

**TIME LIMIT: 2 seconds**

**MEMORY LIMIT: 64 MB**

**PROBLEM NAME: solder**

**INPUT FORMAT:**

**\* Line 1: A single integer: N**

**\* Lines 2..N: Two space-separated integers describing an edge: A and B**

**SAMPLE INPUT (file solder.in):**

**6**

**1 2**

**1 3**

**1 4**

**1 5**

**1 6**

**OUTPUT FORMAT:**

**\* Line 1: A single integer, the cost of soldering the tree together.**

**Note that this number may not always fit in a 32-bit integer.**

**SAMPLE OUTPUT (file solder.out):**

**7**

**OUTPUT DETAILS:**

**Since all nodes in the structure are connected to node 1, we only need to**

**buy one wire of length 2 and three of length 1, for a total cost of 2 \* 2 +**

**1 \* 1 + 1 \* 1 + 1 \* 1 = 7.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***