GOLD PROBLEMS

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

Three problems numbered 1 through 3

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

Problem 1: Fair Shuttle [Russ Cox and Hal Burch, 2006]

Although Farmer John has no problems walking around the fair to

collect prizes or see the shows, his cows are not in such good

shape; a full day of walking around the fair leaves them exhausted.

To help them enjoy the fair, FJ has arranged for a shuttle truck

to take the cows from place to place in the fairgrounds.

FJ couldn't afford a really great shuttle, so the shuttle he rented

traverses its route only once (!) and makes N (1 <= N <= 20,000)

stops (conveniently numbered 1..N) along its path. A total of K (1

<= K <= 50,000) groups of cows conveniently numbered 1..K wish to

use the shuttle, each of the M\_i (1 <= M\_i <= N) cows in group i

wanting to ride from one stop S\_i (1 <= S\_i < E\_i) to another stop

E\_i (S\_i < E\_i <= N) farther along the route.

The shuttle might not be able to pick up an entire group of cows

(since it has limited capacity) but can pick up partial groups as

appropriate.

Given the capacity C (1 <= C <= 100) of the shuttle truck and the

descriptions of the groups of cows that want to visit various sites

at the fair, determine the maximum number of cows that can ride the

shuttle during the fair.

PROBLEM NAME: shuttle

INPUT FORMAT:

\* Line 1: Three space-separated integers: K, N, and C

\* Lines 2..K+1: Line i+1 describes group i of the cows with three

space-separated integers: S\_i, E\_i, and M\_i

SAMPLE INPUT (file shuttle.in):

8 15 3

1 5 2

13 14 1

5 8 3

8 14 2

14 15 1

9 12 1

12 15 2

4 6 1

OUTPUT FORMAT:

\* Line 1: The maximum number of cows that can ride the shuttle at the

fair.

SAMPLE OUTPUT (file shuttle.out):

10

OUTPUT DETAILS:

The shuttle can take 2 cows from stop 1 to stop 5, 3 from 5 to 8, 2 from 8

to 14, 1 from 9 to 12, 1 from 13 to 14, and 1 from 14 to 15.

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

Problem 2: Stock Market [Rob Kolstad(?), 2008]

Despite their innate prudence, the cows took a beating in the home

mortgage market and now are trying their hand at stocks. Happily,

Bessie is prescient and knows not only today's S (2 <= S <= 50)

stock prices but also the future stock prices for a total of D days

(2 <= D <= 10).

Given the matrix of current and future stock prices on various days

(1 <= PR\_sd <= 1,000) and an initial M (1 <= M <= 200,000) units

of money, determine an optimal buying and selling strategy in order

to maximize the gain realized by selling stock on the final day.

Shares must be purchased in integer multiples, and you need not

spend all the money (or any money). It is guaranteed that you will

not be able to earn a profit of more than 500,000 units of money.

Consider the example below of a bull (i.e., improving) market, the

kind Bessie likes most. In this case, S=2 stocks and D=3 days. The

cows have 10 units of money to invest.

Today's price

| Tomorrow's price

| | Two days hence

Stock | | |

1 10 15 15

2 13 11 20

If money is to be made, the cows must purchase stock 1 on day 1.

Selling stock 1 on day 2 and quickly buying stock 2 yields 4 money

in the bank and one share of 2. Selling stock 2 on the final day

brings in 20 money for a total of 24 money when the 20 is added to

the bank.

PROBLEM NAME: stock

INPUT FORMAT:

\* Line 1: Three space-separated integers: S, D, and M

\* Lines 2..S+1: Line s+1 contains the D prices for stock s on days

1..D: PR\_sd

SAMPLE INPUT (file stock.in):

2 3 10

10 15 15

13 11 20

OUTPUT FORMAT:

\* Line 1: The maximum amount of money possible to have after selling

on day D.

SAMPLE OUTPUT (file stock.out):

24

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

Problem 3: Revamping Trails [Percy Liang, 2008]

Farmer John dutifully checks on the cows every day. He traverses

some of the M (1 <= M <= 50,000) trails conveniently numbered 1..M

from pasture 1 all the way out to pasture N (a journey which is

always possible for trail maps given in the test data). The N (1

<= N <= 10,000) pastures conveniently numbered 1..N on Farmer John's

farm are currently connected by bidirectional dirt trails. Each

trail i connects pastures P1\_i and P2\_i (1 <= P1\_i <= N; 1 <= P2\_i

<= N) and requires T\_i (1 <= T\_i <= 1,000,000) units of time to

traverse.

He wants to revamp some of the trails on his farm to save time on

his long journey. Specifically, he will choose K (1 <= K <= 20)

trails to turn into highways, which will effectively reduce the

trail's traversal time to 0. Help FJ decide which trails to revamp

to minimize the resulting time of getting from pasture 1 to N.

TIME LIMIT: 2 seconds

PROBLEM NAME: revamp

INPUT FORMAT:

\* Line 1: Three space-separated integers: N, M, and K

\* Lines 2..M+1: Line i+1 describes trail i with three space-separated

integers: P1\_i, P2\_i, and T\_i

SAMPLE INPUT (file revamp.in):

4 4 1

1 2 10

2 4 10

1 3 1

3 4 100

OUTPUT FORMAT:

\* Line 1: The length of the shortest path after revamping no more than

K edges

SAMPLE OUTPUT (file revamp.out):

1

OUTPUT DETAILS:

K is 1; revamp trail 3->4 to take time 0 instead of 100. The new shortest

path is 1->3->4, total traversal time now 1.

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