Matrix Chain Multiplication (MCM)

MCM: The Problem

When multiply matrices together we perform some number of operations.

MCM: The Problem

When multiply matrices together we perform some number of operations.

- To multiply two matrices together the first matrix has to have the same number of columns as the second matrices's rows.

MCM: The Problem

When multiply matrices together we perform some number of operations.

- To multiply two matrices together the first matrix has to have the same number of columns as the second matrices's rows.
- The resulting matrix has the same number of rows as the first, and the same number of columns as the second.

MCM: The Problem

When multiply matrices together we perform some number of operations.

- To multiply two matrices together the first matrix has to have the same number of columns as the second matrices's rows.
- The resulting matrix has the same number of rows as the first, and the same number of columns as the second.
- For matrices that are rXs and sXt the number of operations to multiply them together is r*s*t.

MCM: The Problem

When multiply matrices together we perform some number of operations.

- To multiply two matrices together the first matrix has to have the same number of columns as the second matrices's rows.
- The resulting matrix has the same number of rows as the first, and the same number of columns as the second.
- For matrices that are rXs and sXt the number of operations to multiply them together is r*s*t.
- Thes resulting matrix would be rXt.

MCM: The Problem

When multiply matrices together we perform some number of operations.

- To multiply two matrices together the first matrix has to have the same number of columns as the second matrices's rows.
- The resulting matrix has the same number of rows as the first, and the same number of columns as the second.
- For matrices that are rXs and sXt the number of operations to multiply them together is r*s*t.
- Thes resulting matrix would be rXt.

We want to multiply a group of matrices together such that the number of operations is minimized.

MCM: Example

We have matrices A, B, and C.

MCM: Example

We have matrices A, B, and C.

- A is 1x5
- B is 5x2
- C is 2x7

MCM: Example

We have matrices A, B, and C

- A is 1x5
- B is 5x2
- C is 2x7

The matrix ABC can be computed either by doing A(BC) or (AB)C.

MCM: Example

We have matrices A, B, and C

- A is 1x5
- B is 5x2
- C is 2x7

The matrix ABC can be computed either by doing A(BC) or (AB)C.

- A(BC) # ops = (5*2*7) + (1*5*7)

MCM: Example

We have matrices A, B, and C

- A is 1x5
- B is 5x2
- C is 2x7

The matrix ABC can be computed either by doing A(BC) or (AB)C.

- A(BC) # ops = (5*2*7) + (1*5*7) = 70 + 35 = 105

MCM: Example

We have matrices A, B, and C

- A is 1x5
- B is 5x2
- C is 2x7

The matrix ABC can be computed either by doing A(BC) or (AB)C.

- A(BC) # ops = (5*2*7) + (1*5*7) = 70 + 35 = 105
- (AB)C # ops = (1*5*2) + (1*2*7)

MCM: Example

We have matrices A, B, and C

- A is 1x5
- B is 5x2
- C is 2x7

The matrix ABC can be computed either by doing A(BC) or (AB)C.

- A(BC) # ops = (5*2*7) + (1*5*7) = 70 + 35 = 105
- (AB)C # ops = (1*5*2) + (1*2*7) = 10 + 14 = 24

MCM: Example

We have matrices A, B, and C

- A is 1x5
- B is 5x2
- C is 2x7

The matrix ABC can be computed either by doing A(BC) or (AB)C.

- A(BC) # ops = (5*2*7) + (1*5*7) = 70 + 35 = 105
- (AB)C # ops = (1*5*2) + (1*2*7) = 10 + 14 = 24

This can be solved using a DP.

MCM: Example

We have matrices A, B, and C

- A is 1x5
- B is 5x2
- C is 2x7

The matrix ABC can be computed either by doing A(BC) or (AB)C.

- A(BC) # ops = (5*2*7) + (1*5*7) = 70 + 35 = 105
- (AB)C # ops = (1*5*2) + (1*2*7) = 10 + 14 = 24

This can be solved using a DP. Let's work on a more fun example.

MCM Problems

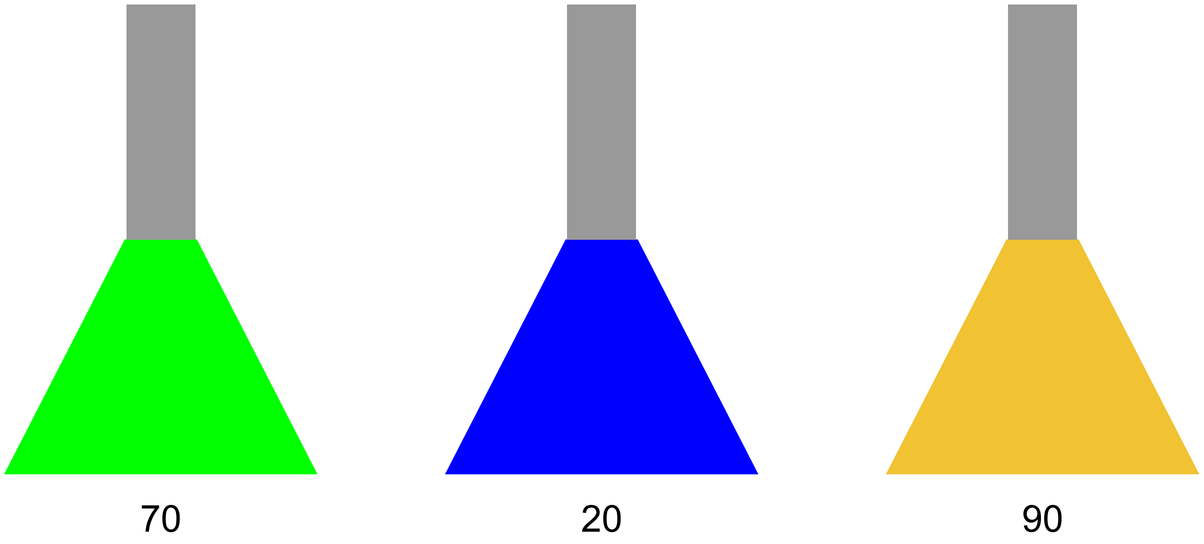
https://www.spoj.com/problems/MIXTURES/

https://www.interviewbit.com/problems/rod-cutting/

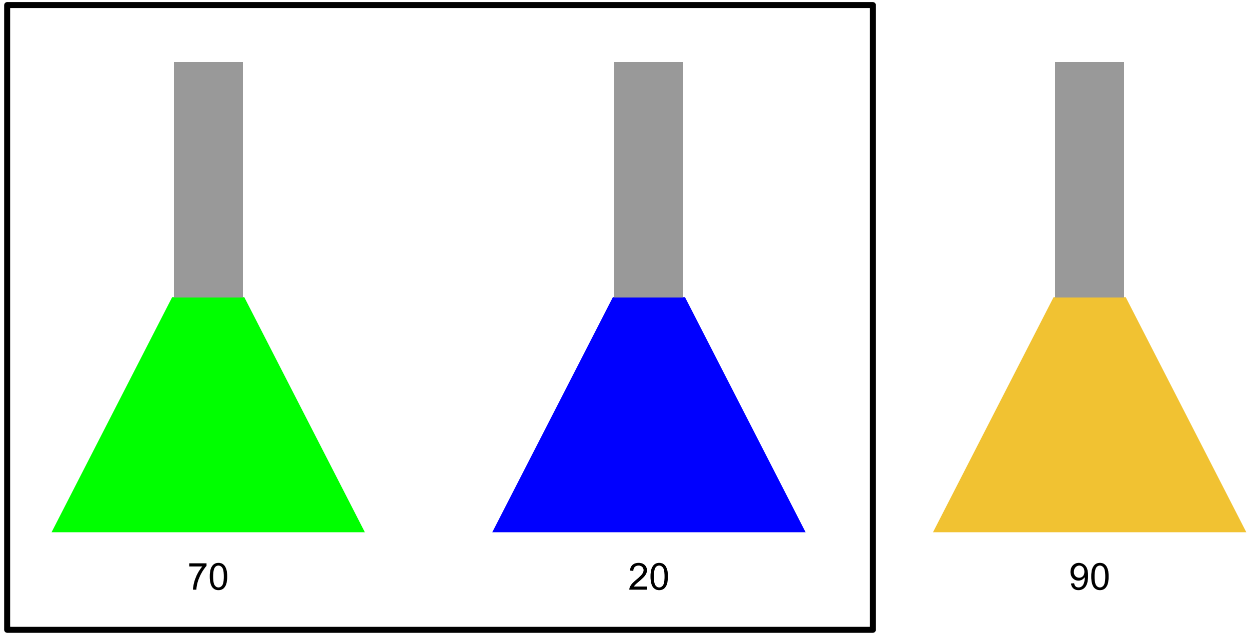
Problem (SPOJ)

Harry Potter is going to mix n potions together which are currently line up. The potions have some integer value from 0 to 99. Only adjacent potions can be mixed. Potions that are mixed go back into place in a combined container. Potions of value a and b produce an amount of smoke equal to a*b. The new combined mixture has a value of (a+b)%100. Find the minimum amount of smoke that can be produced after having combined all the potions.

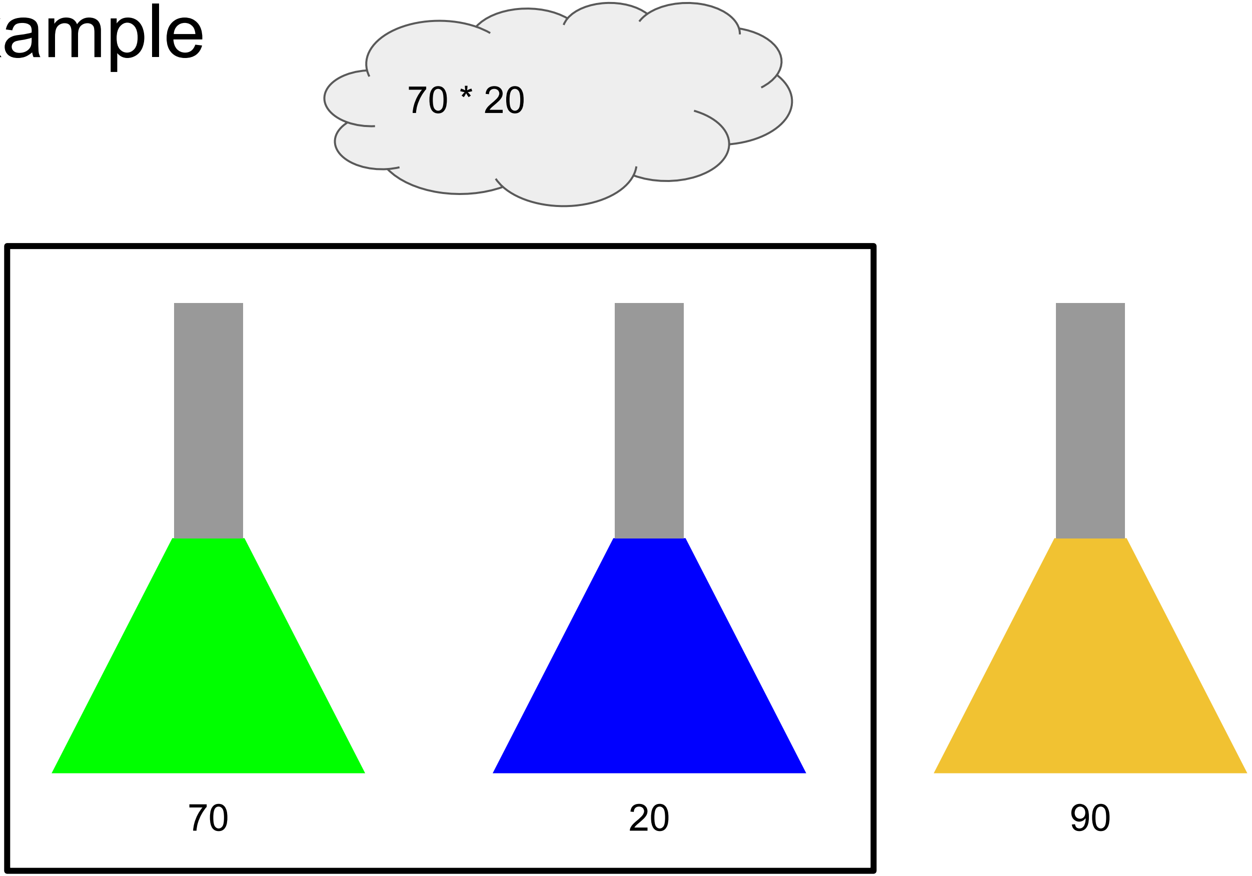
Example



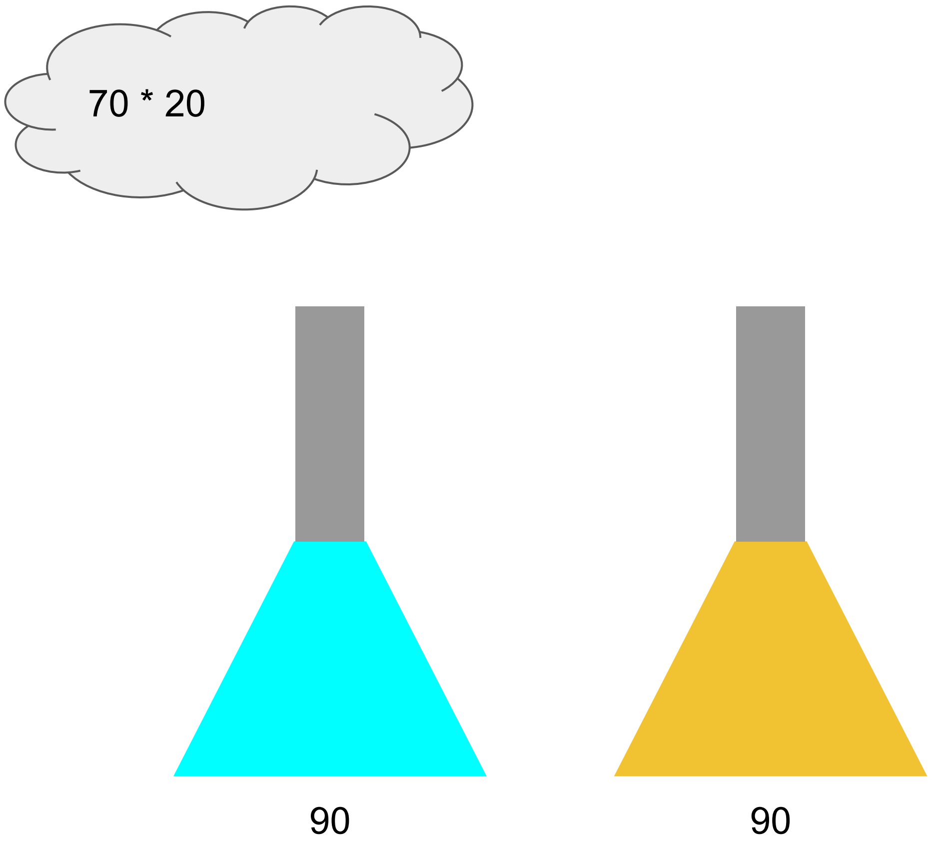
Example



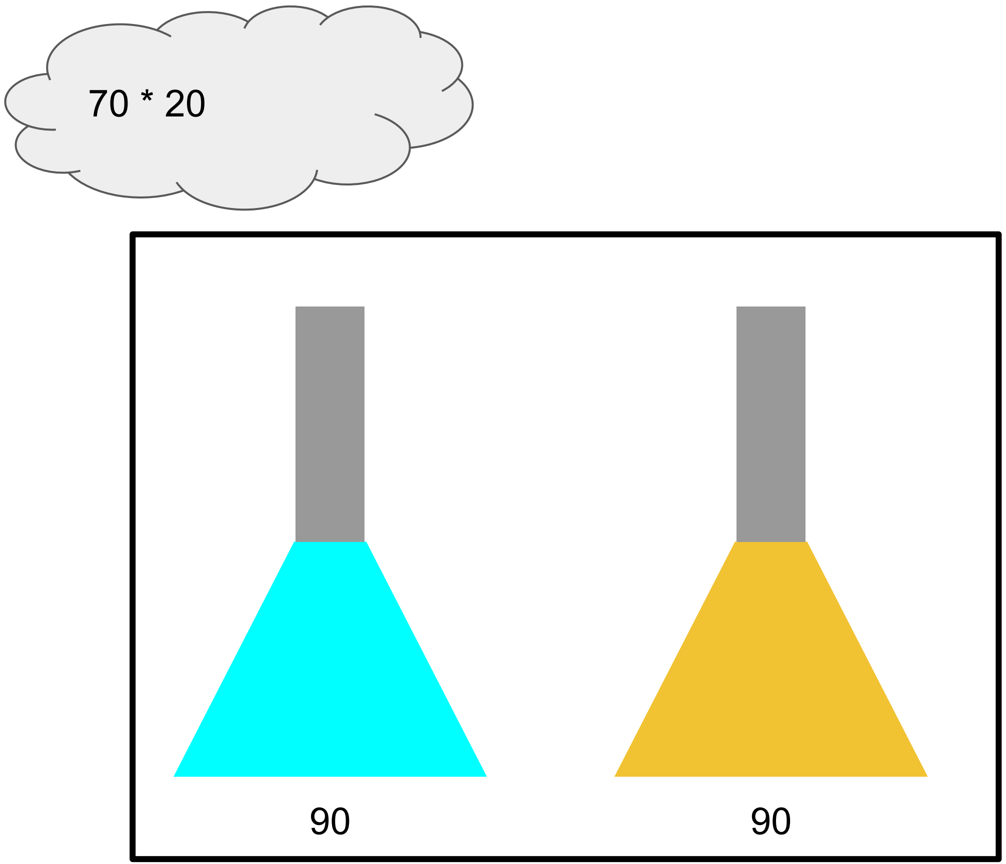
Example



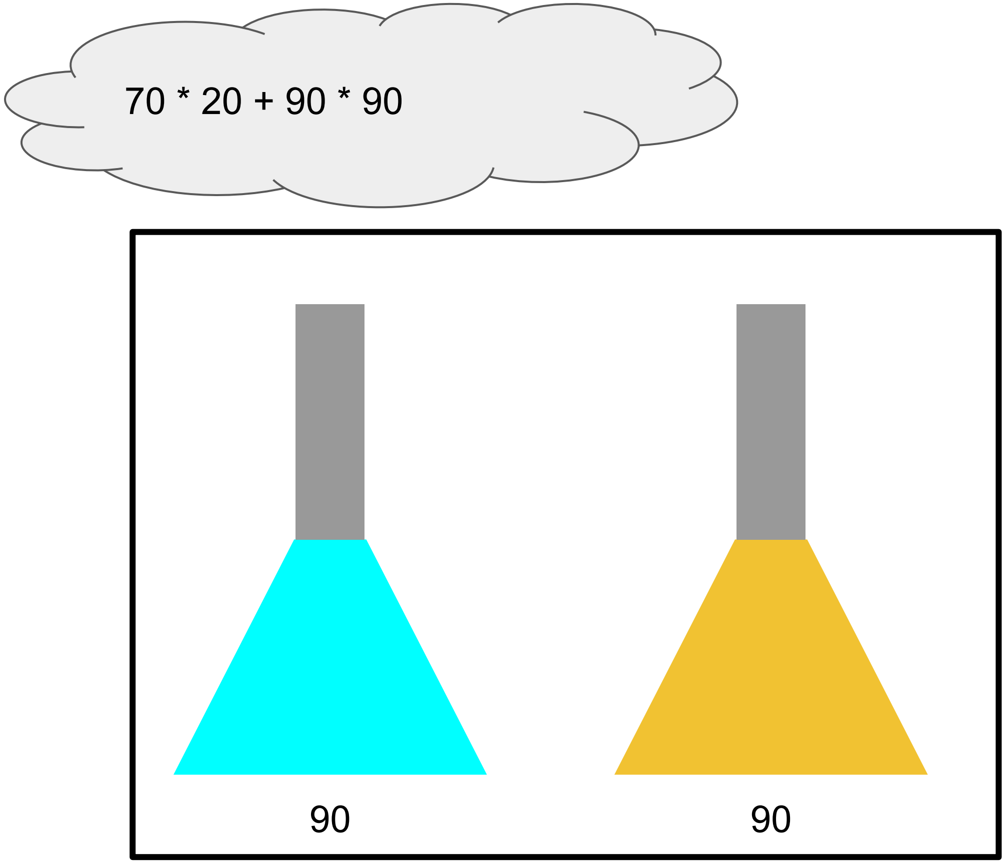
Example



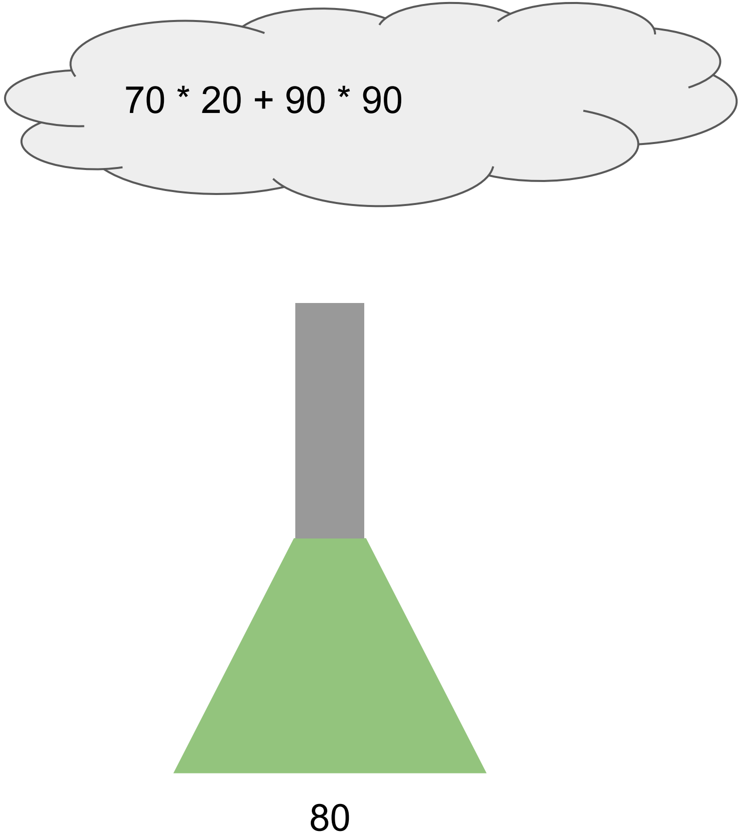
Example



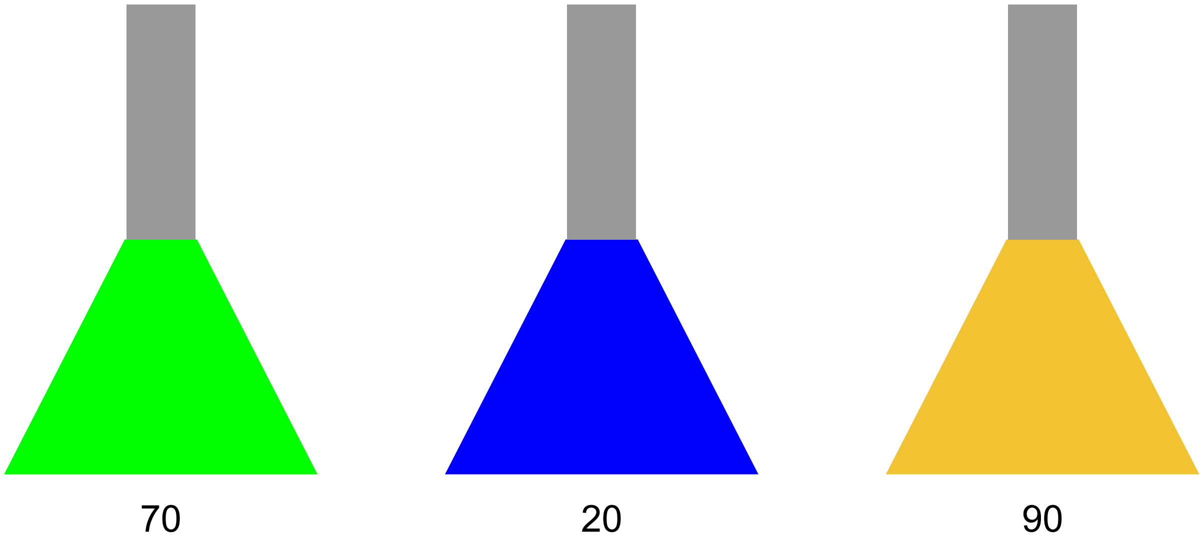
Example



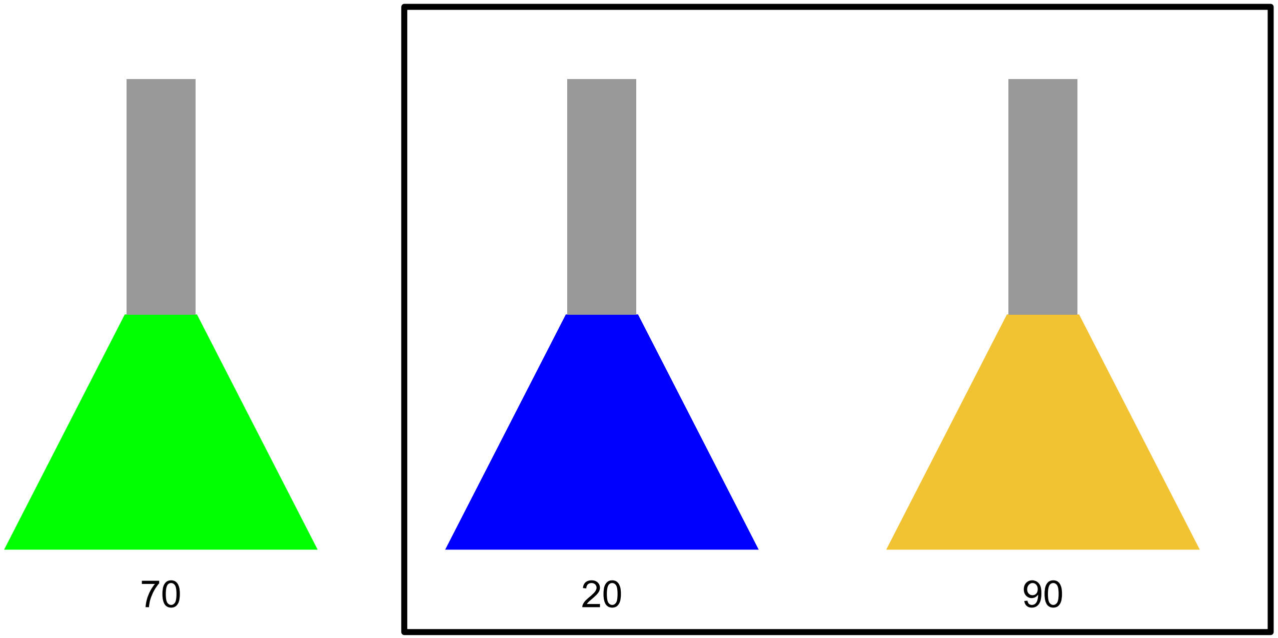
Example



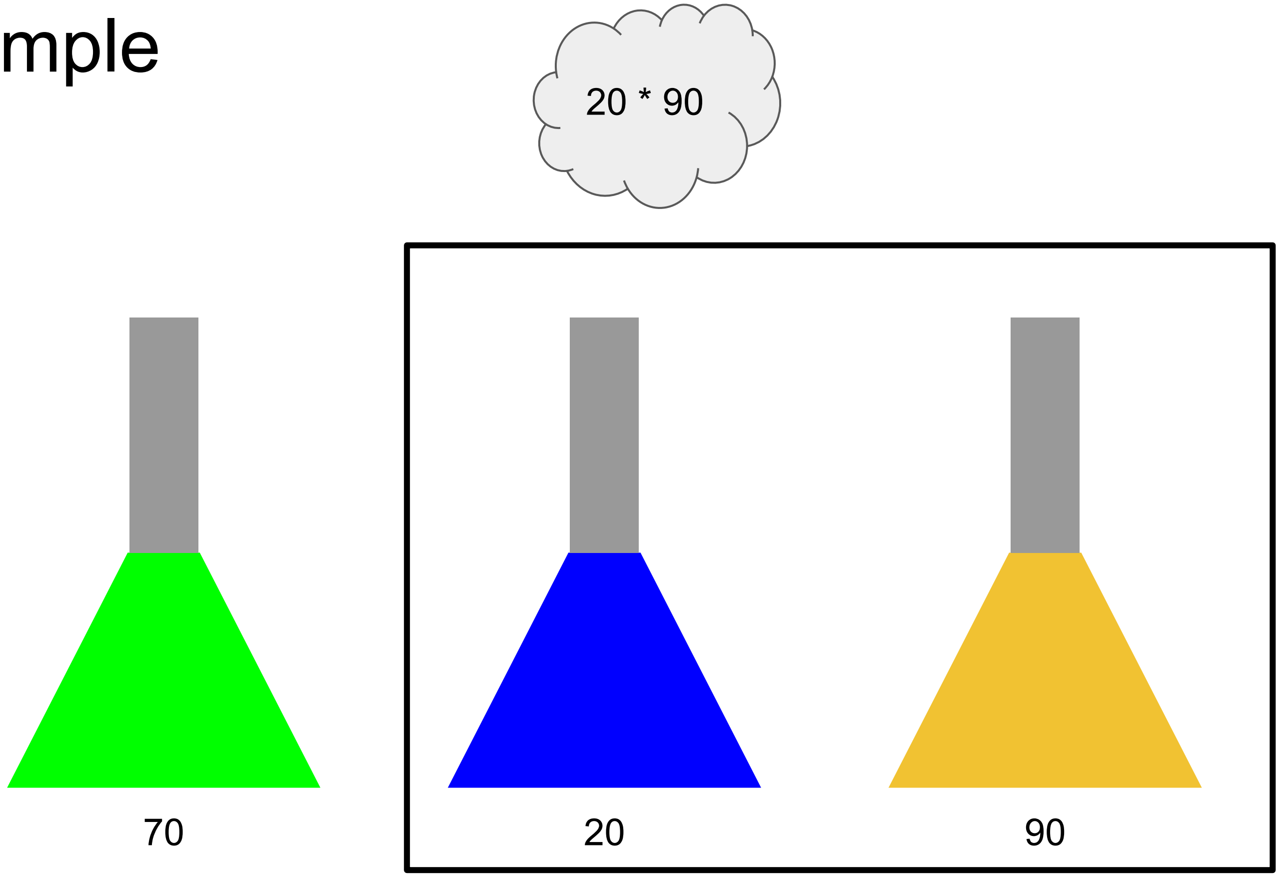
Example



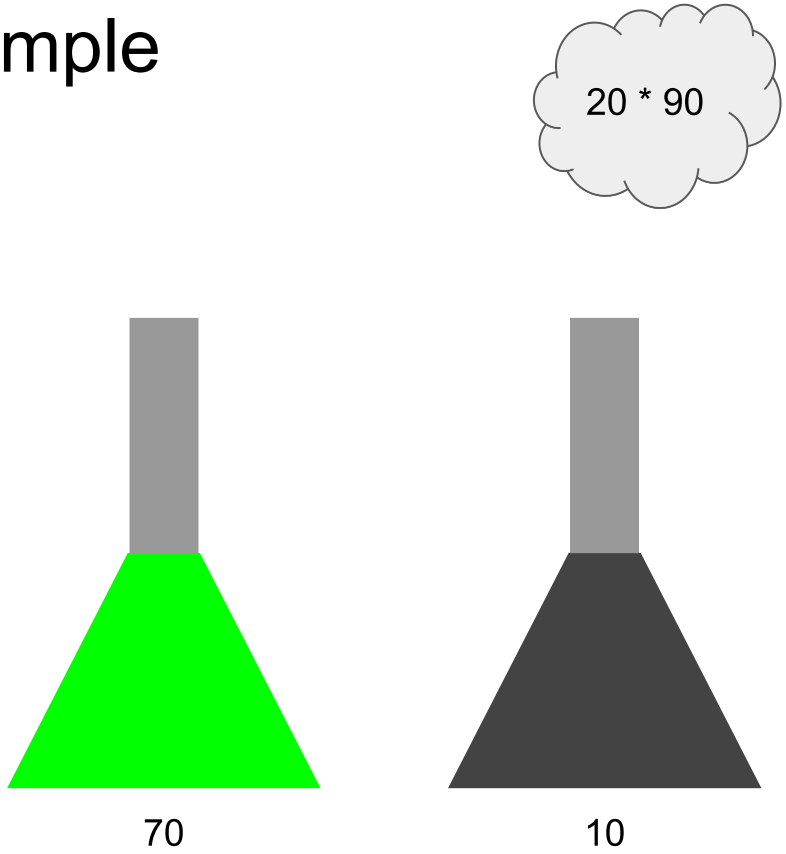
Example



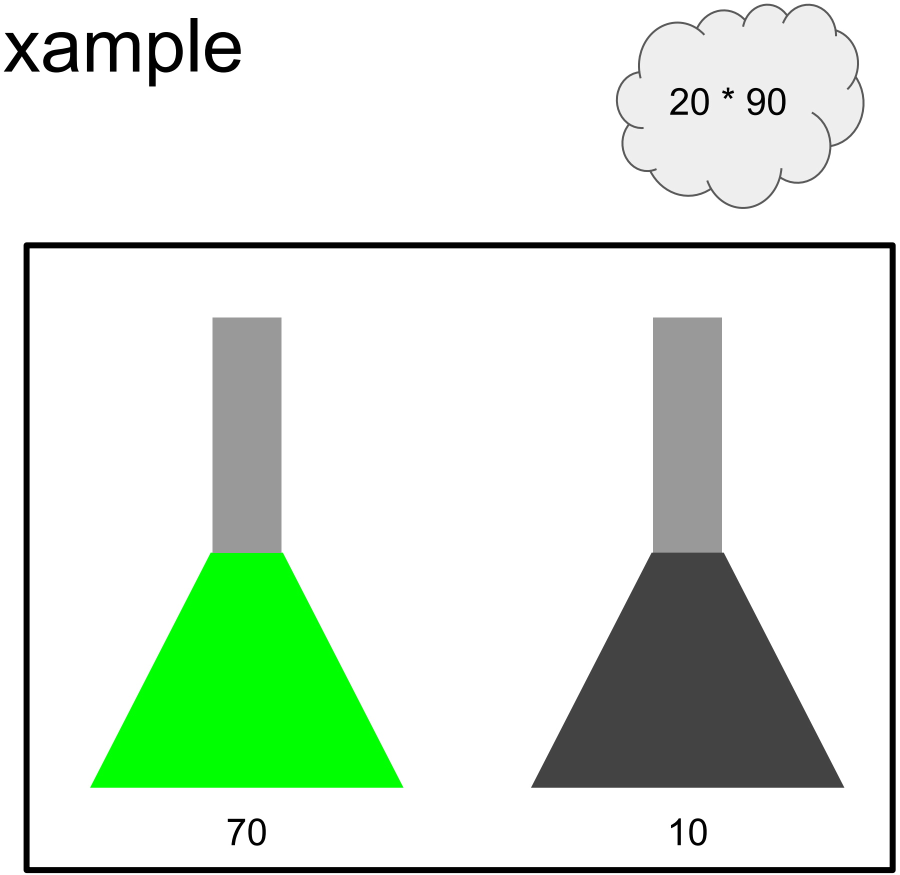
Example



Example

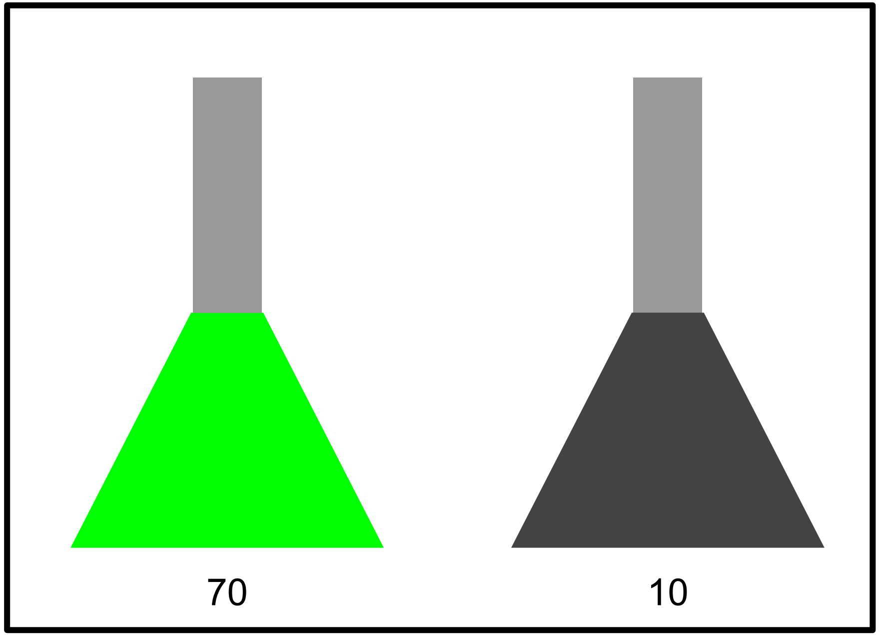


Example



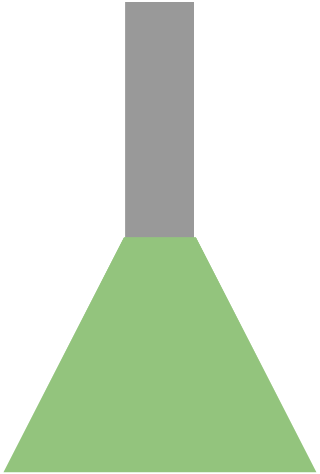
Example

20 * 90 + 70 * 10



Example

20 * 90 + 70 * 10



80

Idea

We could merge all potions using all possible combinations at each level using recursion

Idea

We could merge all potions using all possible combinations at each level using recursion

N-1 possible values in the first level

Idea

We could merge all potions using all possible combinations at each level using recursion

N-1 possible values in the first level

N-2 possible values in the second level

Idea

We could merge all potions using all possible combinations at each level using recursion

N-1 possible values in the first level

N-2 possible values in the second level

And so on...

Idea

We could merge all potions using all possible combinations at each level using recursion

N-1 possible values in the first level

N-2 possible values in the second level

And so on…

O((N-1)!)

Idea

We could merge all potions using all possible combinations at each level using recursion

N-1 possible values in the first level

N-2 possible values in the second level

And so on…

O((N-1)!)

Harry has 100 potions...

Better Idea

Observation a mixture at any given time comes from a contiguous subsequence of potions from the original set. (Can’t skip a potion)

Better Idea

Observation a mixture at any given time comes from a contiguous subsequence of potions from the original set. (Can’t skip a potion)

A mixture will always have the same value regardless of ordering of potion combination.

Better Idea

Observation a mixture at any given time comes from a contiguous subsequence of potions from the original set. (Can’t skip a potion)

A mixture will always have the same value regardless of ordering of potion combination.

We know that the last potion created is done so by merging two potions that perfectly split the original set of potions.

Better Idea

Observation a mixture at any given time comes from a contiguous subsequence of potions from the original set. (Can’t skip a potion)

A mixture will always have the same value regardless of ordering of potion combination.

We know that the last potion created is done so by merging two potions that perfectly split the original set of potions.

That last merger (given a specific split) will always produce the same amount of smoke.

Better Idea

Observation a mixture at any given time comes from a contiguous subsequence of potions from the original set. (Can't skip a potion)

A mixture will always have the same value regardless of ordering of potion combination.

We know that the last potion created is done so by merging two potions that perfectly split the original set of potions.

That last merger (given a specific split) will always produce the same amount of smoke.

$A_1 + A_2 + A_3 + \ldots + A_k$ AND $A_{k+1} + A_{k+2} + \ldots + A_N$

Which Means...

If we could figure out the cheapest method to get 1 to K.

Which Means...

If we could figure out the cheapest method to get 1 to K.

And the cheapest method for getting K+1 to N we could answer the question faster.

Which Means...

If we could figure out the cheapest method to get 1 to K.

And the cheapest method for getting K+1 to N we could answer the question faster.

Runtime is a little better.

Which Means...

If we could figure out the cheapest method to get 1 to K.

And the cheapest method for getting K+1 to N we could answer the question faster.

Runtime is a little better.

$O(C_{N-1})$

Which Means...

If we could figure out the cheapest method to get 1 to K.

And the cheapest method for getting K+1 to N we could answer the question faster.

Runtime is a little better.

$O(C_{N-1})$

When N is 100

Which Means...

If we could figure out the cheapest method to get 1 to K.

And the cheapest method for getting K+1 to N we could answer the question faster.

Runtime is a little better.

$O(C_{N-1})$

When N is 100

896519947090131496687170070074100632420837521538745909320

Which Means...

If we could figure out the cheapest method to get 1 to K.

And the cheapest method for getting K+1 to N we could answer the question faster.

Runtime is a little better.

$O(C_{N-1})$

When N is 100

896519947090131496687170070074100632420837521538745909320

Just a tad over 10^8

Duplicate Work

But we compute segments multiple times

Duplicate Work

But we compute segments multiple times

Once a segment from a to b is computed store and reuse it

Duplicate Work

But we compute segments multiple times

Once a segment from a to b is computed store and reuse it

Called the Matrix Chain Multiplication DP

Duplicate Work

But we compute segments multiple times

Once a segment from a to b is computed store and reuse it

Called the Matrix Chain Multiplication DP

B/C It can be done for determine number of operations when multiply matrices

Problem

We want to multiply matrices A, B, C, …, and X together.

Problem

We want to multiply matrices A, B, C, …, and X together.

Matrices can be thought of as 2D grids.

Problem

We want to multiply matrices A, B, C, …, and X together.

Matrices can be thought of as 2D grids.

When multiplying A (r x c) and B (n x m) together we must have the middle dimensions equal

Problem

We want to multiply matrices A, B, C, …, and X together.

Matrices can be thought of as 2D grids.

When multiplying A (r x c) and B (n x m) together we must have the middle dimensions equal, so c must be equal to n.

Problem

We want to multiply matrices A, B, C, …, and X together.

Matrices can be thought of as 2D grids.

When multiplying A (r x c) and B (n x m) together we must have the middle dimensions equal, so c must be equal to n. The effort to multiply A and B is rcm (mcr?) (e.g. outside_1 inside outside_2).

Problem

We want to multiply matrices A, B, C, …, and X together.

Matrices can be thought of as 2D grids.

When multiplying A (r x c) and B (n x m) together we must have the middle dimensions equal, so c must be equal to n. The effort to multiply A and B is rcm (mcr?) (e.g. outside_1 inside outside_2). The result has dimension (r x m).

Problem

We want to multiply matrices A, B, C, …, and X together.

Matrices can be thought of as 2D grids.

When multiplying A (r x c) and B (n x m) together we must have the middle dimensions equal, so c must be equal to n. The effort to multiply A and B is rcm (mcr?) (e.g. outside_1 inside outside_2). The result has dimension (r x m).

Multiplication is associative

Problem

We want to multiply matrices A, B, C, …, and X together.

Matrices can be thought of as 2D grids.

When multiplying A (r x c) and B (n x m) together we must have the middle dimensions equal, so c must be equal to n. The effort to multiply A and B is rcm (mcr?) (e.g. outside_1 inside outside_2). The result has dimension (r x m).

Multiplication is associative, so A(BC) = (AB)C.

Problem

We want to multiply matrices A, B, C, …, and X together.

Matrices can be thought of as 2D grids.

When multiplying A (r x c) and B (n x m) together we must have the middle dimensions equal, so c must be equal to n. The effort to multiply A and B is rcm (mcr?) (e.g. outside_1 inside outside_2). The result has dimension (r x m).

Multiplication is associative, so A(BC) = (AB)C.

The question is what is the least effort to find ABCD...X?

Solution

Find the split.

Solution

Find the split.

(ABC...K)(LMN...X)

Solution

Find the split.

(ABC...K)(LMN...X)

Solve both sides and add in the effort for the final multiplication

Solution

Find the split.

(ABC...K)(LMN...X)

Solve both sides and add in the effort for the final multiplication (rows of A) times (columns of K OR rows of L) times (columns of X).

Solution

Find the split.

(ABC...K)(LMN...X)

Solve both sides and add in the effort for the final multiplication (rows of A) times (columns of K OR rows of L) times (columns of X).

Do that harry potter DP!