Longest common subsequence

INPUT: two strings

OUTPUT: longest common subsequence

ACTGAACTCTGTGCACT

TGACTCAGCACAAAAAC

Longest common subsequence

INPUT: two strings

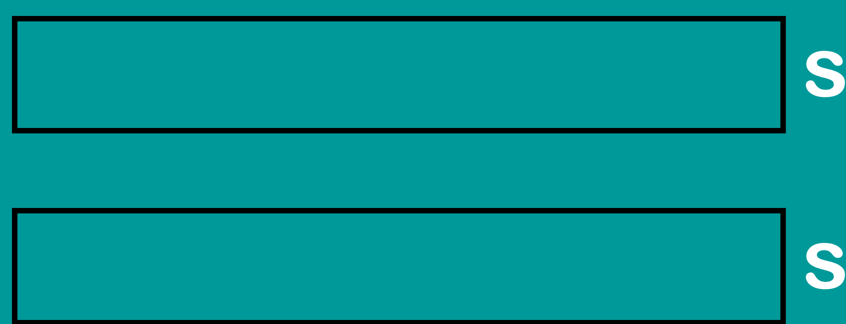
OUTPUT: longest common subsequence

ACTGAACTCTGTGCACT

TGACTCAGCACAAAAAC

Longest common subsequence

If the sequences end with the same symbol s, then LCS ends with s.



Longest common subsequence

Sequences $x_1,\ldots,x_n$, and $y_1,\ldots,y_m$

LCS(i,j) = length of a longest common subsequence of $x_1,\ldots,x_i$ and $y_1,\ldots,y_j$

Longest common subsequence

Sequences $x_1,\ldots,x_n$, and $y_1,\ldots,y_m$

$LCS(i,j)$ = length of a longest common subsequence of $x_1,\ldots,x_i$ and $y_1,\ldots,y_j$

if $x_i = y_j$ then

$LCS(i,j) =$

Longest common subsequence

Sequences $x_1,\ldots,x_n$, and $y_1,\ldots,y_m$

LCS(i,j) = length of a longest common subsequence of $x_1,\ldots,x_i$ and $y_1,\ldots,y_j$

if $x_i = y_j$ then

$$LCS(i,j) = 1 + LCS(i-1,j-1)$$

Longest common subsequence

Sequences $x_1,\ldots,x_n$, and $y_1,\ldots,y_m$

LCS(i,j) = length of a longest common subsequence of $x_1,\ldots,x_i$ and $y_1,\ldots,y_j$

if $x_i \neq y_j$ then

$$LCS(i,j) = \max(LCS(i-1,j), LCS(i,j-1))$$

x_i and y_j cannot both be in LCS

Longest common subsequence

Sequences $x_1,\ldots,x_n$, and $y_1,\ldots,y_m$

$LCS(i,j)$ = length of a longest common subsequence of $x_1,\ldots,x_i$ and $y_1,\ldots,y_j$

if $x_i = y_j$ then
$LCS(i,j) = 1 + LCS(i-1,j-1)$

if $x_i \neq y_j$ then
$LCS(i,j) = \max(LCS(i-1,j),LCS(i,j-1))$

Longest common subsequence

Running time ?

Sequences $x_1,\ldots,x_n$, and $y_1,\ldots,y_m$

LCS(i,j) = length of a longest common subsequence of $x_1,\ldots,x_i$ and $y_1,\ldots,y_j$

if $x_i = y_j$ then
LCS(i,j) = 1 + LCS(i-1,j-1)

if $x_i \neq y_j$ then
LCS(i,j) = max (LCS(i-1,j),LCS(i,j-1))

Longest common subsequence

Running time = O(mn)

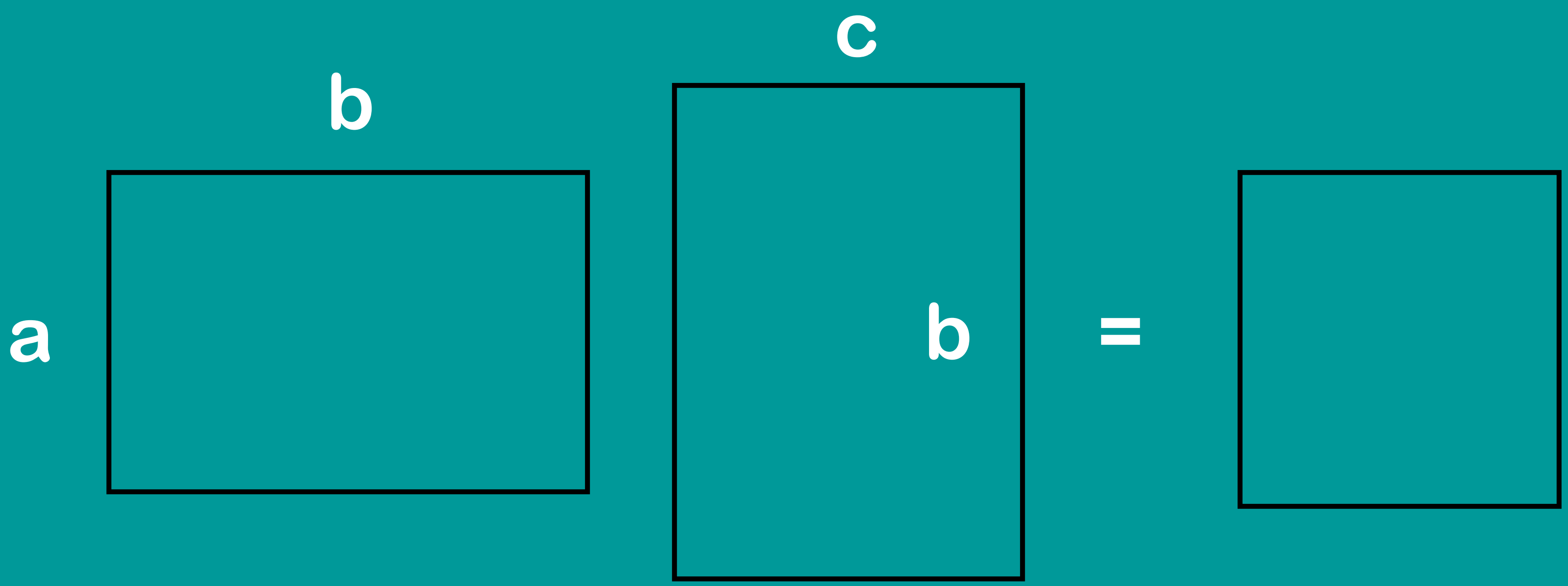
Sequences $x_1,\ldots,x_n$, and $y_1,\ldots,y_m$

LCS(i,j) = length of a longest common subsequence of $x_1,\ldots,x_i$ and $y_1,\ldots,y_j$

if $x_i = y_j$ then
LCS(i,j) = 1 + LCS(i-1,j-1)

if $x_i \neq y_j$ then
LCS(i,j) = max (LCS(i-1,j),LCS(i,j-1))

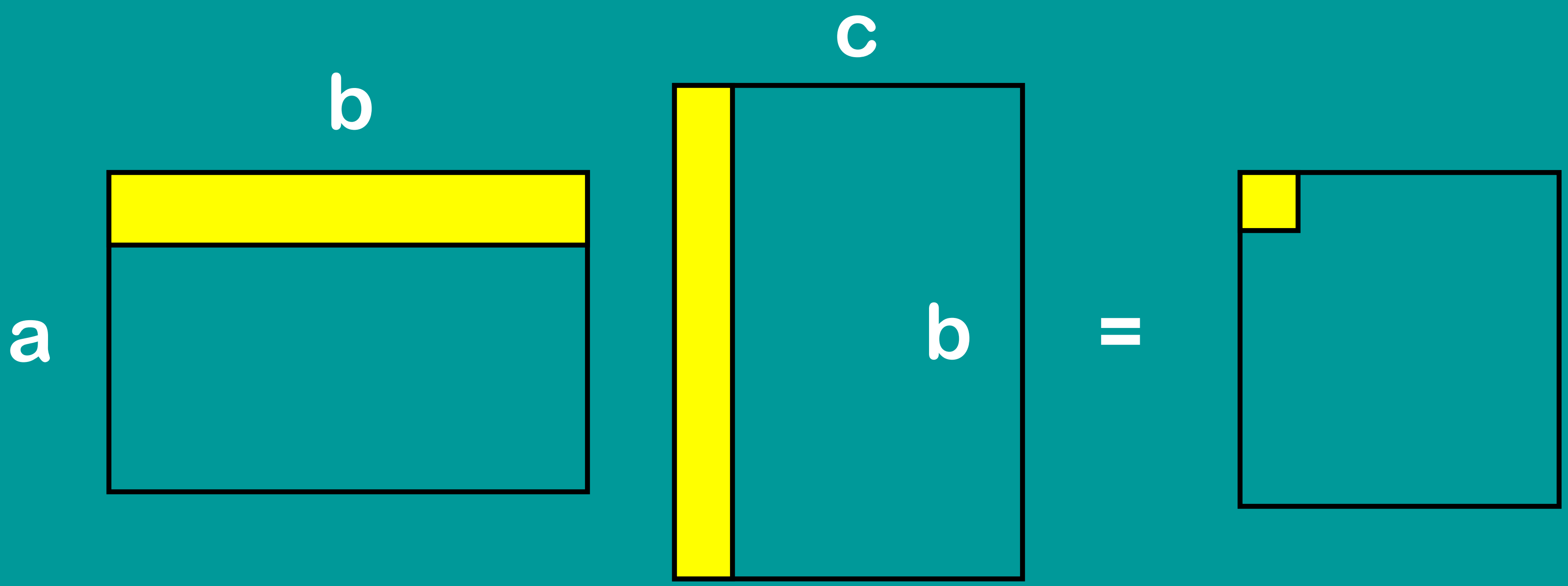
Optimal matrix multiplication



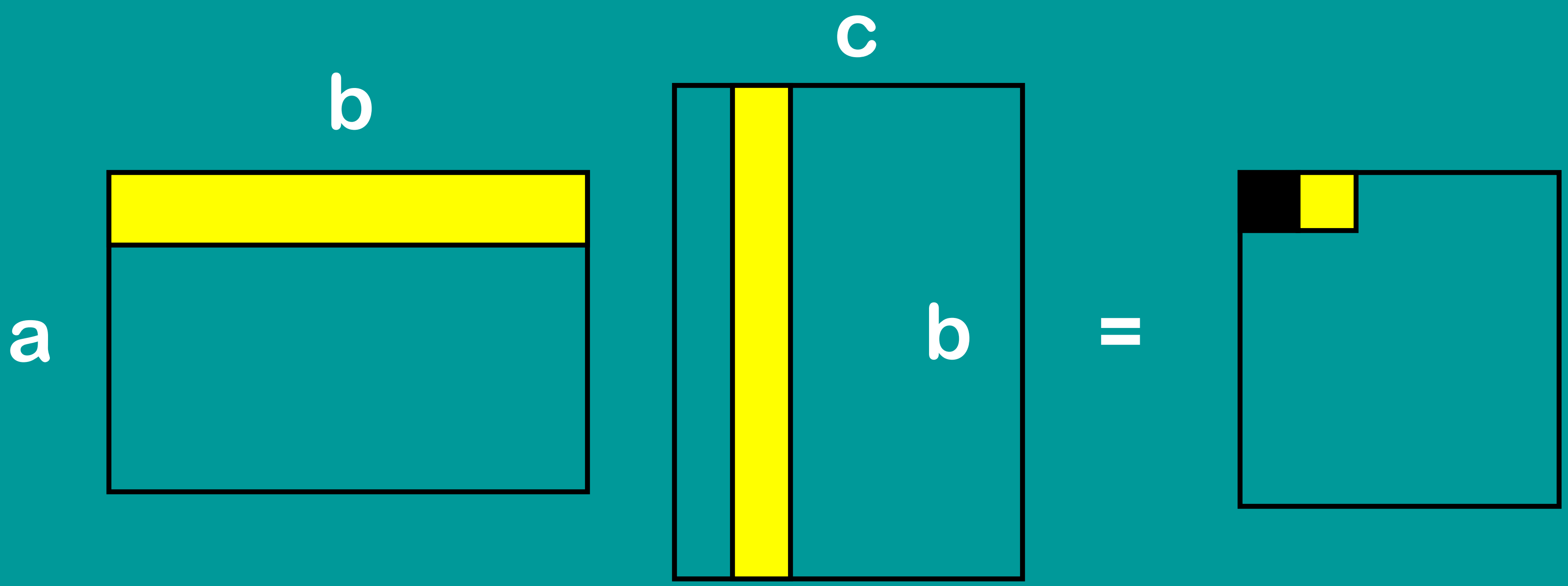
A = a x b matrix
B = b x c matrix

How many operations to compute AB ?

Optimal matrix multiplication



Optimal matrix multiplication



each entry of A * B takes $\Theta(b)$ time

need to compute ac entries $\Rightarrow$ $\Theta(abc)$ time total

Optimal matrix multiplication

N x N
matrix

A

N x N
matrix

B

Compute AB, time = ?

Optimal matrix multiplication

N x N matrix

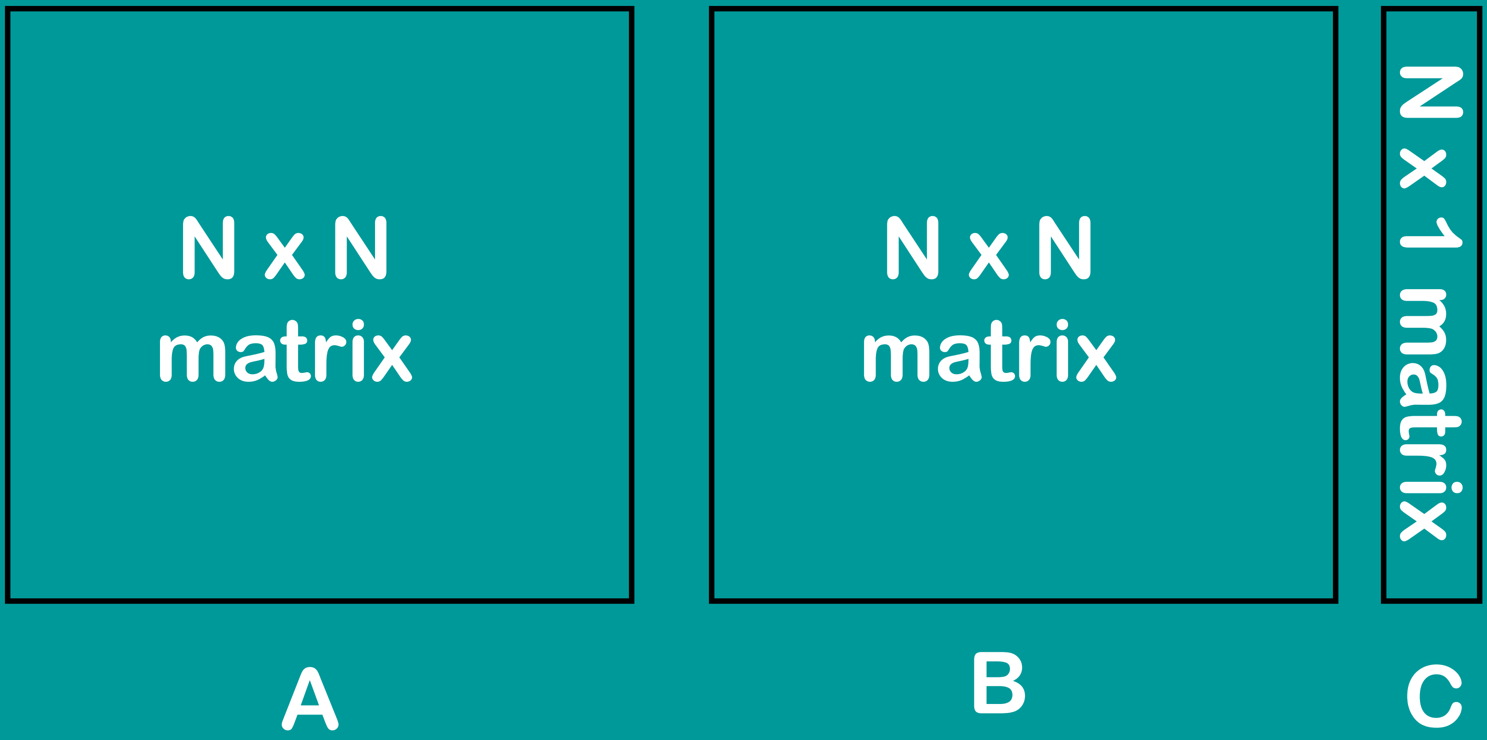
N x N matrix

A

B

Compute AB, time = $\Theta(N^3)$

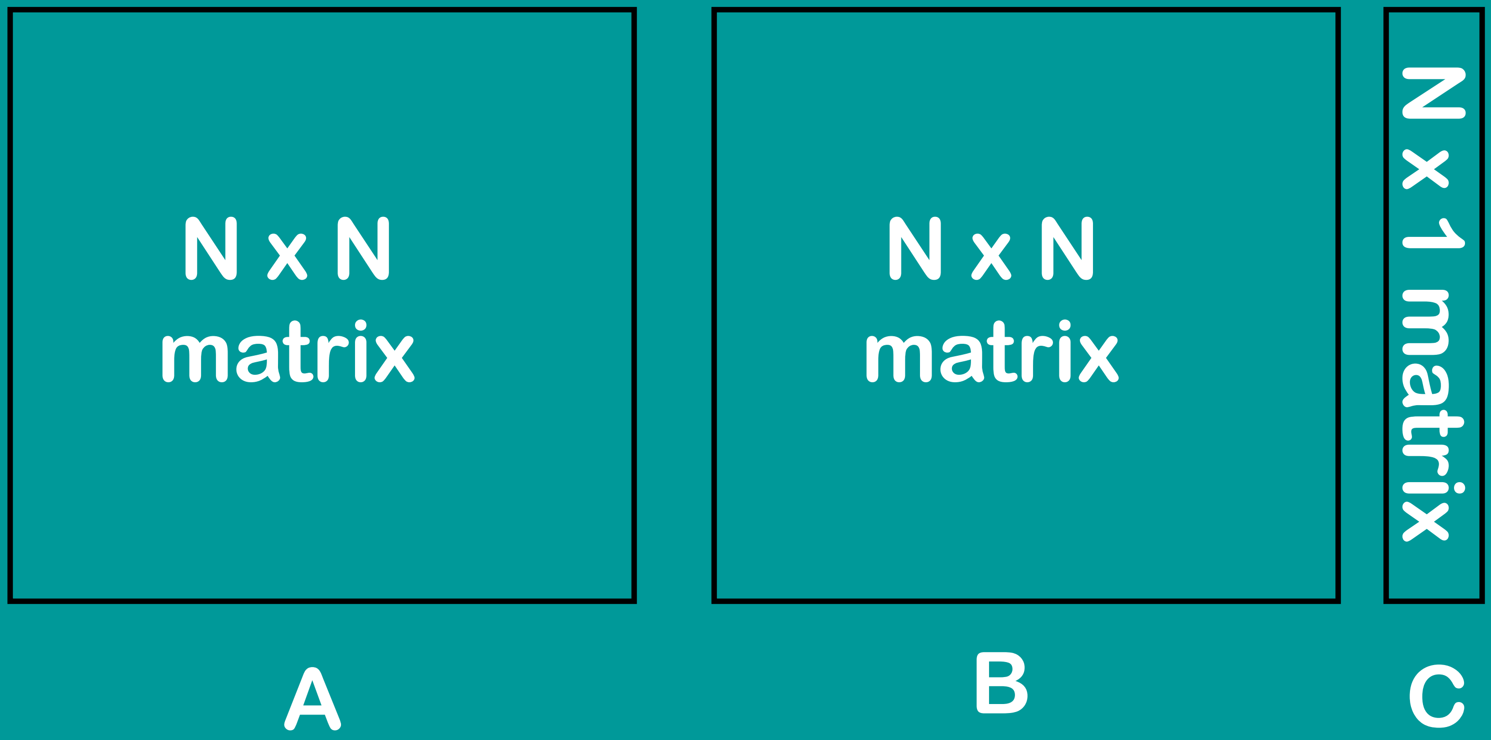
Optimal matrix multiplication



Compute AB, time = $\Theta(N^3)$

Compute ABC, time = ?

Optimal matrix multiplication

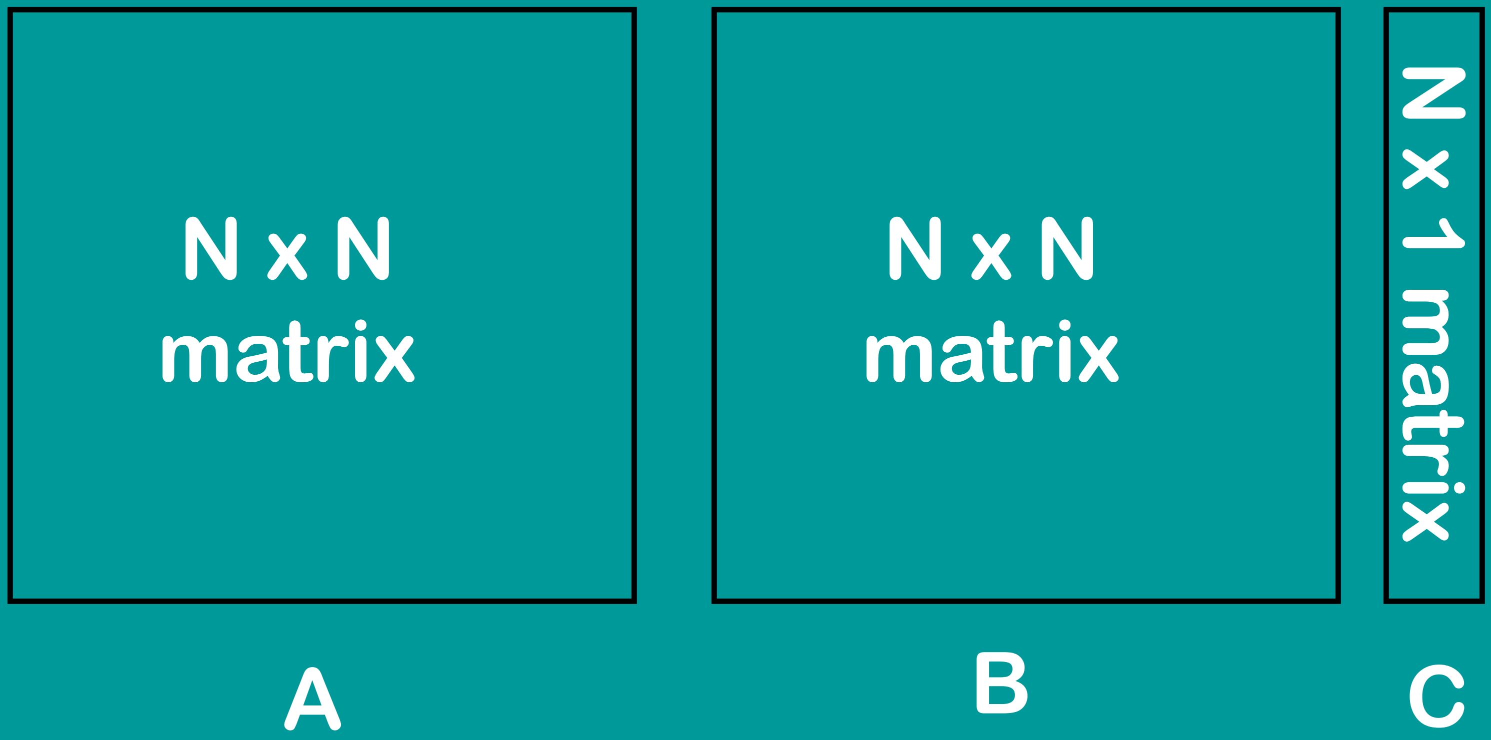


Compute D=BC, time = ?

Compute AD, time = ?

Compute ABC, time = ?

Optimal matrix multiplication

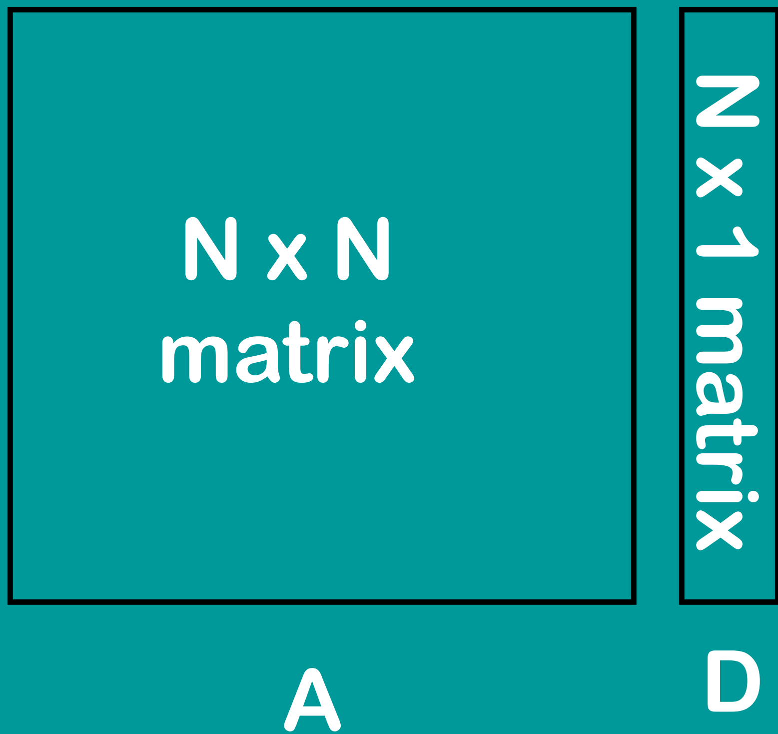


Compute D=BC, time = $\Theta(N^2)$

Compute AD, time = ?

Compute ABC, time = ?

Optimal matrix multiplication

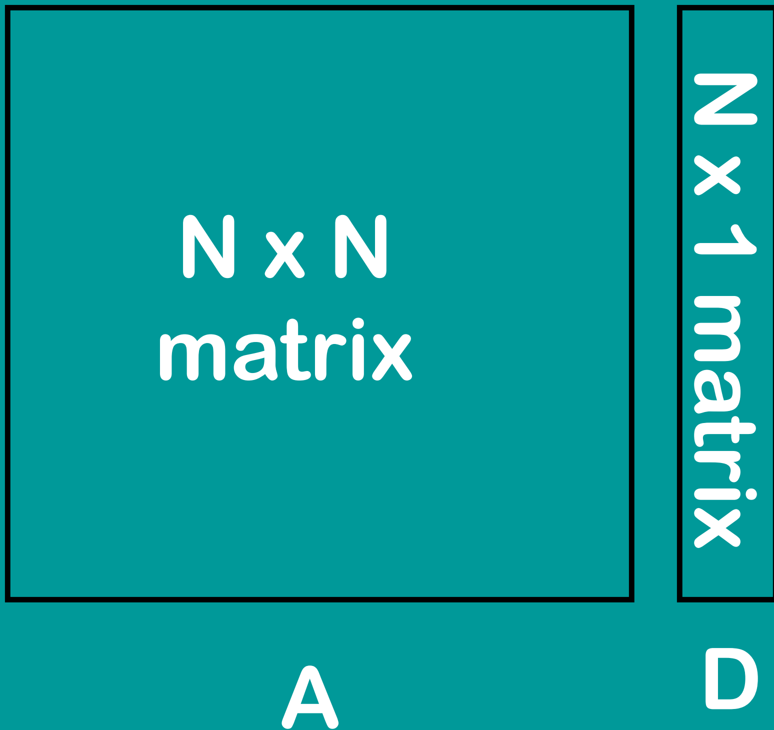


Compute D=BC, time = $\Theta(N^2)$

Compute AD, time = ?

Compute ABC, time = ?

Optimal matrix multiplication

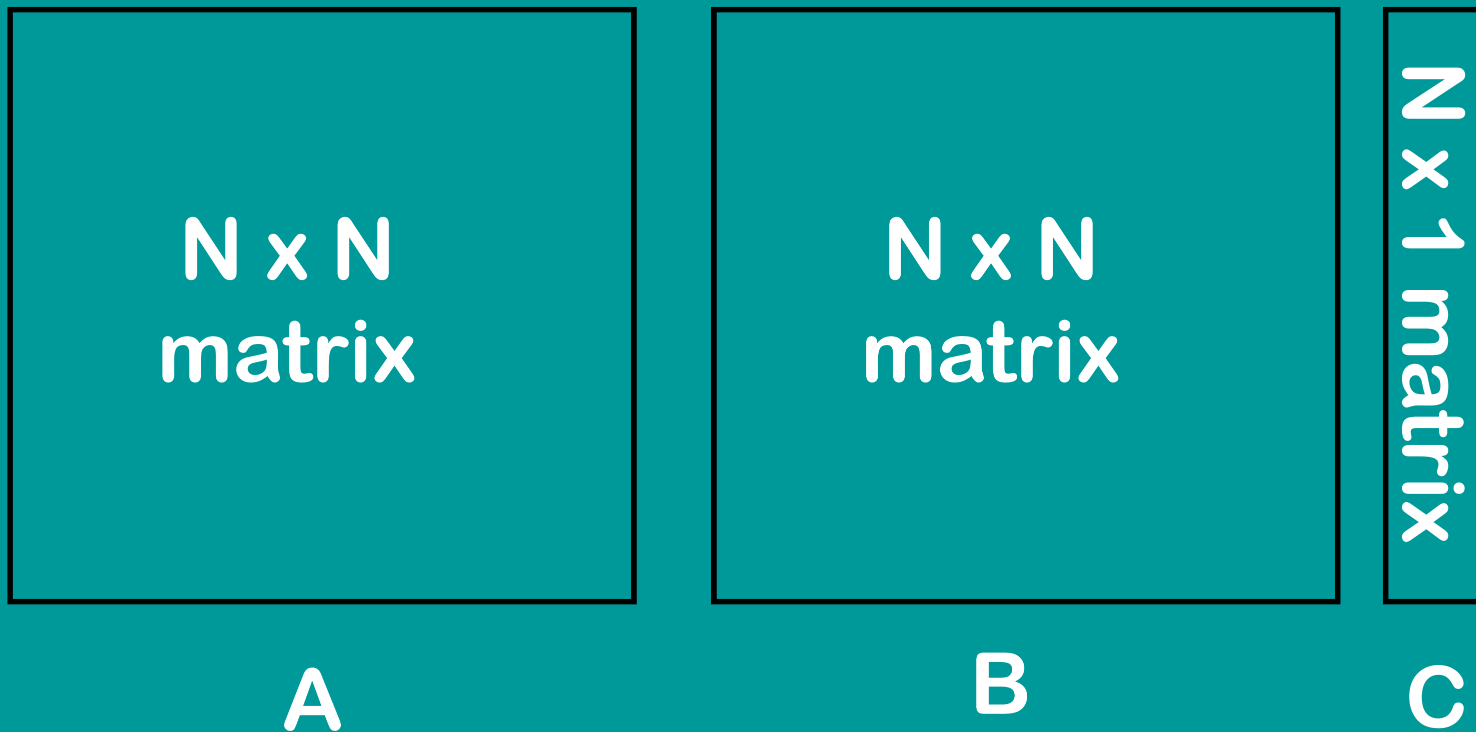


Compute D=BC, time = $\Theta(N^2)$

Compute AD, time = $\Theta(N^2)$

Compute ABC, time = $\Theta(N^2)$

Optimal matrix multiplication



$$(AB)C = ABC = A(BC)$$

The order of evaluation

does not change the result

can change the amount of work needed

Optimal matrix multiplication

$a_1, a_2, a_3, \ldots, a_n$

n-1 matrices of sizes

$a_1 \times a_2$ B_1

$a_2 \times a_3$ B_2

$a_3 \times a_4$ B_3

....

$a_{n-1} \times a_n$ B_{n-1}

What order should we multiply them in?

Optimal matrix multiplication

$B_1\ B_2\ B_3\ B_4\ \ldots\ B_{n-1}$

$B_1\ (B_2\ B_3\ B_4\ \ldots\ B_{n-1}\ )$

$(B_1\ B_2)\ (B_3\ B_4\ \ldots\ B_{n-1}\ )$

$(B_1\ B_2\ B_3)\ (B_4\ \ldots\ B_{n-1}\ )$

$\ldots$

$(B_1\ B_2\ B_3\ B_4\ \ldots)\ B_{n-1}$

Optimal matrix multiplication

$B_1 \; B_2 \; B_3 \; B_4 \; \ldots \; B_{n-1}$

K[i,j] = the minimal number of operations needed to multiply $B_i \ldots B_j$

$B_i \; (B_{i+1} \; B_{i+2} \; \ldots \; B_j)$	$K[i,i] + K[i+1,j] + a_i a_{i+1} a_{j+1}$
$(B_i \; B_{i+1}) \; (B_{i+2} \; \ldots \; B_j)$	$K[i,i+1] + K[i+2,j] + a_i a_{i+2} a_{j+1}$
$(B_i \; B_{i+1} \; B_{i+2}) \; (\ldots \; B_j)$	$K[i,i+2] + K[i+3,j] + a_i a_{i+3} a_{j+1}$
$\ldots$	
$(B_1 \; B_2 \; B_3 \; \ldots) \; B_j$	$K[i,j-1] + K[j,j] + a_i a_j a_{j+1}$

Optimal matrix multiplication

$B_1\ B_2\ B_3\ B_4 \ldots B_{n-1}$

$K[i,j]$ = the minimal number of operations needed to multiply $B_i \ldots B_j$

$$K[i,i]=0$$

$$K[i,j] = \min_{i \le w < j} K[i,w] + K[w+1,j] + a_i\, a_{w+1}\, a_j$$

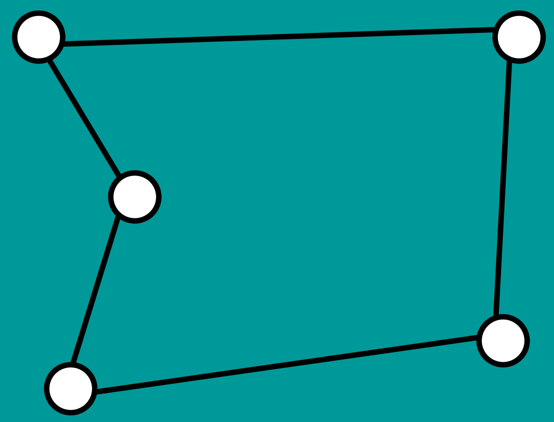
Travelling Salesman Problem

INPUT:

N cities, NxN symmetric matrix D,
D(i,j) = distance between city i and j

OUTPUT:

the shortest tour visiting all the cities



Travelling Salesman Problem

Algorithm 1 – try all possibilities

for each permutation π of $\{1,\ldots,n\}$
visit the cities in the order π,
compute distance travelled,

pick the best solution

running time = ?

Travelling Salesman Problem

Algorithm 1 – try all possibilities

for each permutation π of $\{1,\ldots,n\}$
visit the cities in the order π,
compute distance travelled,

pick the best solution

running time $\approx$ n!

is $(n+1)! = O(n!)$?

Travelling Salesman Problem

for each subset S of the cities with
$|S| \geq 2$ and each $u,v \in S$

K[S,u,v] the length of the shortest path that
* starts at u
* ends at v
* visits all cities in S

How large is K ?

Travelling Salesman Problem

for each subset S of the cities with
$|S| \geq 2$ and each $u,v \in S$

K[S,u,v] the length of the shortest path that

* starts at u
* ends at v
* visits all cities in S

How large is K ?

$$\approx 2^n n^2$$

Travelling Salesman Problem

K[S,u,v]

some vertex $w \in S - \{u,v\}$
must be visited first

d(u,w) = we get to w
K[S-u,w,v] = we need to get from w to v
and visit all vertices in S-u

Travelling Salesman Problem

K[S,u,v] the length of the shortest path that

* starts at u
* ends at v
* visits all cities in S

if $S=\{u,v\}$ then $K[S,u,v]=d(u,v)$

if $|S|>2$ then

$$K[S,u,v] = \min_{w \in S-\{u,v\}} K[S-u,w,v] + d(u,w)$$

Travelling Salesman Problem

if $S=\{u,v\}$ then $K[S,u,v]=d(u,v)$

if $|S|>2$ then

$$K[S,u,v] = \min_{w \in S-\{u,v\}} K[S-u,w,v] + d(u,w)$$

Running time = ?

$K \approx 2^n n^2$

Travelling Salesman Problem

if $S=\{u,v\}$ then $K[S,u,v]=d(u,v)$

if $|S|>2$ then

$$K[S,u,v] = \min_{w \in S-\{u,v\}} K[S-u,w,v] + d(u,w)$$

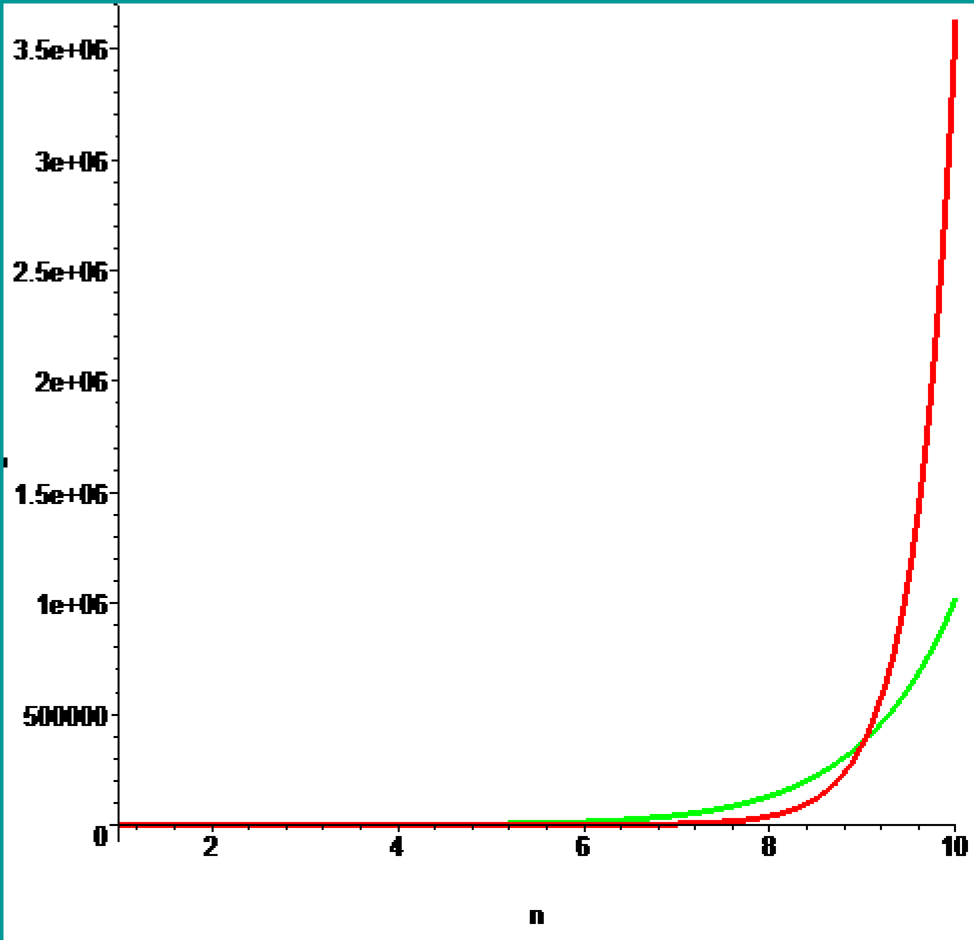
Running time = $O(n^3\, 2^n)$

$K \approx 2^n\, n^2$

Travelling Salesman Problem

dynamic programming = $O(n^3 \, 2^n)$

brute force = $O(n!)$



Longest increasing subsequence

INPUT: numbers $a_1, a_2, \ldots, a_n$

OUTPUT: longest increasing subsequence

1,9,2,4,7,5,6

1,9,**2**,**4**,7,**5**,**6**

Longest increasing subsequence

INPUT: numbers $a_1, a_2, \ldots, a_n$

OUTPUT: longest increasing subsequence

reduce to a problem that we saw today

Longest increasing subsequence

INPUT: numbers $a_1, a_2, \ldots, a_n$

OUTPUT: longest increasing subsequence

Longest increasing subsequence

INPUT: numbers $a_1, a_2, \dots, a_n$

OUTPUT: longest increasing subsequence

K[0..n,0..n]

K[i,j] = the minimum last element of an increasing seqence in $a_1, \dots, a_i$ of length j (if no sequence $\Rightarrow \infty$)

Longest increasing subsequence

K[0..n,0..n]

K[i,j] = the minimum last element of an increasing seqence in $a_1, \dots, a_i$ of length j (if no sequence $\Rightarrow \infty$)

true/false: $K[i,j] \leq K[i,j+1]$?

Longest increasing subsequence

K[0..n,0..n]

K[i,j] = the minimum last element of an increasing seqence in $a_1, \ldots, a_i$ of length j (if no sequence $\Rightarrow \infty$)

K[0,j] = ? for $j \geq 1$

K[0,0] = ?

Longest increasing subsequence

K[0..n,0..n]

K[i,j] = the minimum last element of an increasing seqence in $a_1, \ldots ,a_i$ of length j (if no sequence $\Rightarrow \infty$)

$K[0,j] = \infty$ for $j \geq 1$

$K[0,0] = -\infty$

Longest increasing subsequence

K[0..n,0..n]

K[i,j] = the minimum last element of an increasing seqence in $a_1, \ldots, a_i$ of length j (if no sequence $\Rightarrow \infty$)

K[i,j] = ?

Longest increasing subsequence

$K[0..n,0..n]$

$K[i,j]$ = the minimum last element of an increasing seqence in $a_1, \ldots ,a_i$ of length j (if no sequence $\Rightarrow \infty$)

$K[i,j] = a_i$ if $a_i < K[i-1,j]$ and $a_i > K[i-1,j-1]$

$K[i,j] = K[i-1,j]$ otherwise

Longest increasing subsequence

$K[0..n,0..n]$

$K[i,j]$ = the minimum last element of an increasing seqence in $a_1, \ldots ,a_i$ of length j (if no sequence $\Rightarrow \infty$)

$K[i,0] = -\infty$
$K[i,1] =$
$K[i,2] =$
$\ldots$
$K[i,j] =$
$K[i,j+1] = \infty$

$a_i < K[i-1,j]$
and
$a_i > K[i-1,j-1]$

Longest increasing subsequence

$K[0,0] = -\infty$
$K[0,1] = \infty$
$K[0,2] = \infty$
$K[0,3] = \infty$
$K[0,4] = \infty$
$K[0,5] = \infty$
$K[0,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[1,0] = -\infty$
$K[1,1] = 1$
$K[1,2] = \infty$
$K[1,3] = \infty$
$K[1,4] = \infty$
$K[1,5] = \infty$
$K[1,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[1,0] = -\infty$
$K[1,1] = 1$
$K[1,2] = \infty$
$K[1,3] = \infty$
$K[1,4] = \infty$
$K[1,5] = \infty$
$K[1,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[2,0] = -\infty$
$K[2,1] = 1$
$K[2,2] = 9$
$K[2,3] = \infty$
$K[2,4] = \infty$
$K[2,5] = \infty$
$K[2,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[2,0] = -\infty$
$K[2,1] = 1$
$K[2,2] = 9$
$K[2,3] = \infty$
$K[2,4] = \infty$
$K[2,5] = \infty$
$K[2,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[3,0] = -\infty$
$K[3,1] = 1$
$K[3,2] = 2$
$K[3,3] = \infty$
$K[3,4] = \infty$
$K[3,5] = \infty$
$K[3,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[3,0] = -\infty$
$K[3,1] = 1$
$K[3,2] = 2$
$K[3,3] = \infty$
$K[3,4] = \infty$
$K[3,5] = \infty$
$K[3,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[4,0] = -\infty$
$K[4,1] = 1$
$K[4,2] = 2$
$K[4,3] = 4$
$K[4,4] = \infty$
$K[4,5] = \infty$
$K[4,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[4,0] = -\infty$
$K[4,1] = 1$
$K[4,2] = 2$
$K[4,3] = 4$
$K[4,4] = \infty$
$K[4,5] = \infty$
$K[4,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[5,0] = -\infty$
$K[5,1] = 1$
$K[5,2] = 2$
$K[5,3] = 4$
$K[5,4] = 7$
$K[5,5] = \infty$
$K[5,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[5,0] = -\infty$
$K[5,1] = 1$
$K[5,2] = 2$
$K[5,3] = 4$
$K[5,4] = 7$
$K[5,5] = \infty$
$K[5,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[6,0] = -\infty$
$K[6,1] = 1$
$K[6,2] = 2$
$K[6,3] = 4$
$K[6,4] = 5$
$K[6,5] = \infty$
$K[6,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[6,0] = -\infty$
$K[6,1] = 1$
$K[6,2] = 2$
$K[6,3] = 4$
$K[6,4] = 5$
$K[6,5] = \infty$
$K[6,6] = \infty$

1,9,2,4,7,5,6

Longest increasing subsequence

$K[7,0] = -\infty$
$K[7,1] = 1$
$K[7,2] = 2$
$K[7,3] = 4$
$K[7,4] = 5$
$K[7,5] = 6$
$K[7,6] = \infty$

answer = 5

1,9,2,4,7,5,6