Algorithm Practice 3

Design and analysis of algorithm - CS112.N21.KHTN

Huỳnh Đặng Vĩnh Hiền, 21520029
Lương Toàn Bách, 21521845
Group 5

Falcuty of Computer Science,
University of Information Technology,
Vietnam National Univeristy - Ho Chi Minh City

June 20, 2023

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Table of Contents

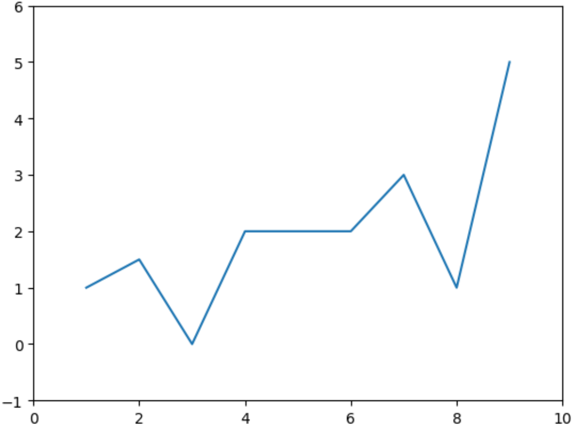
Problem 1: Extrema

Extrema - Introduction

Attetion: The exercise not in here in the wecode.

Extrema - Example

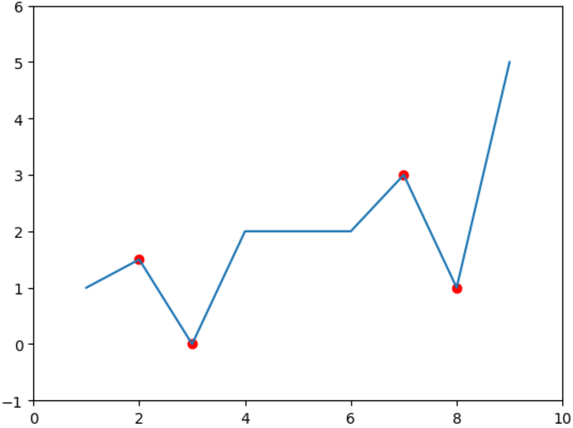
Example: Input: 1 1.5 0 2 2 2 3 1 5



Hình

Extrema - Example

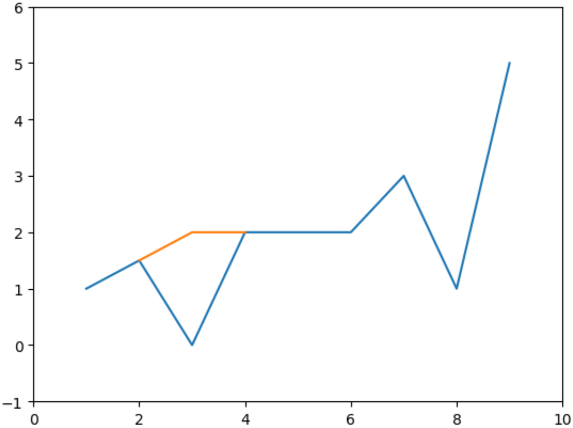
Extrema at the beginning state $\rightarrow$ 4.



Hình

Extrema - Example

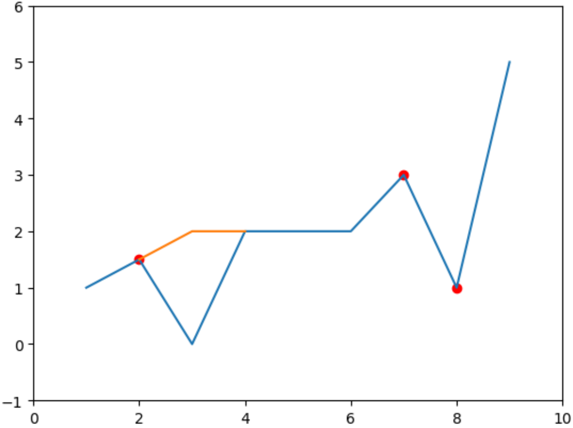
Query: "T 2 2" (T is short for "Temporary")



Hình

Extrema - Example

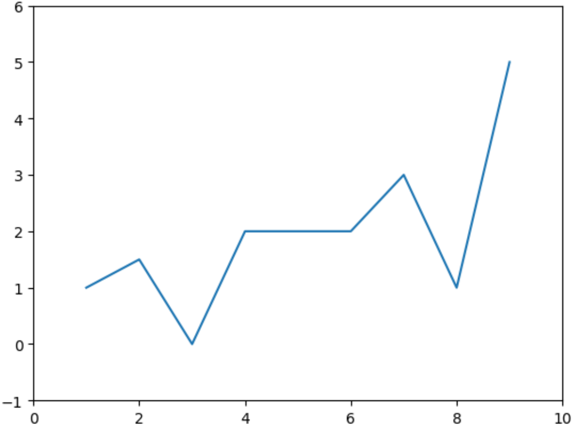
After "T 3 2" query $\rightarrow$ 3 then reverse the value for x_2: $x_2 = 0$



Hình

Extrema - Example

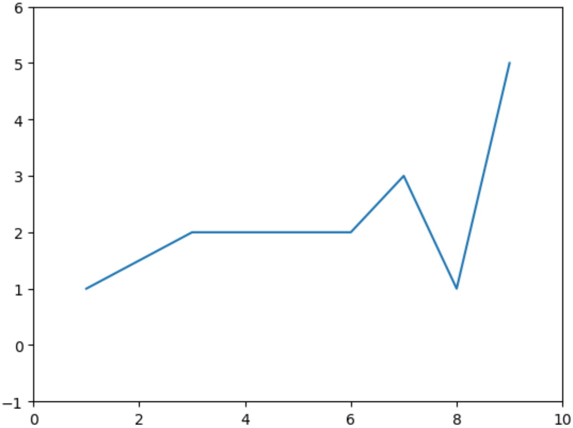
After "T 3 2" query $\rightarrow$ 3 then reverse the value for x_2: $x_2 = 0$



Hình

Extrema - Example

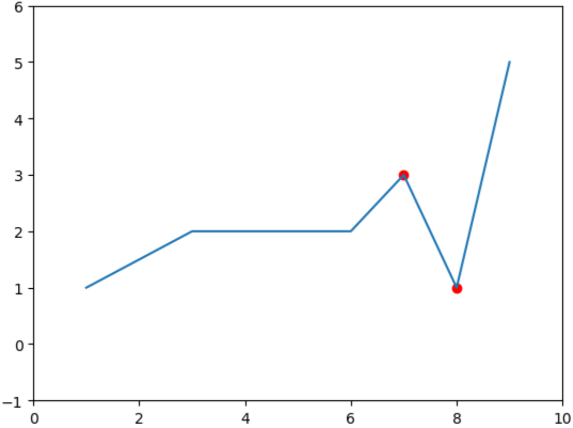
Query: "P 3 2" (P is short for "Permanent")



Hình

Extrema - Example

After "P 3 2" query $\rightarrow$ 2

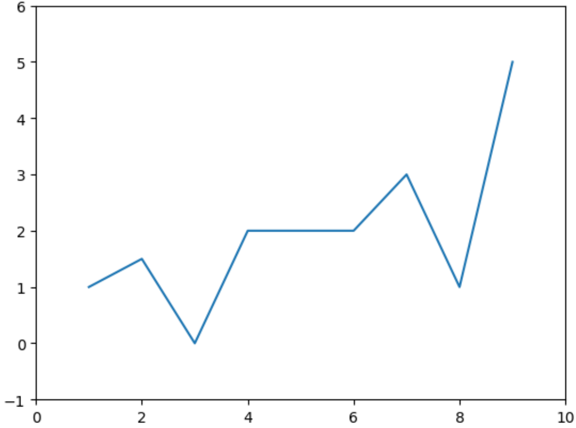


Hình

Extrema - Approach

We can observe that:

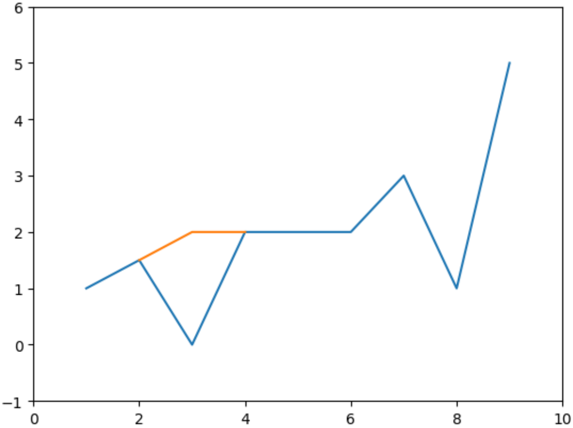
- The first and the last element can't be extrema.



Hình

We can observe that:

- The first and the last element can't be extrema.
- One query can only effect 3 point



Hình

Extrema - Solution

Check whether a_i is a extrema then save the result in an array b_i with all element and calculate beginning number of extremas c.

$$b_i = \mathbb{1}(\text{" } x_i \text{ is a extrema "})\forall\ 1 \le i \le n, \quad c = \sum_{i=2}^{n-1} b_i$$

For each query:

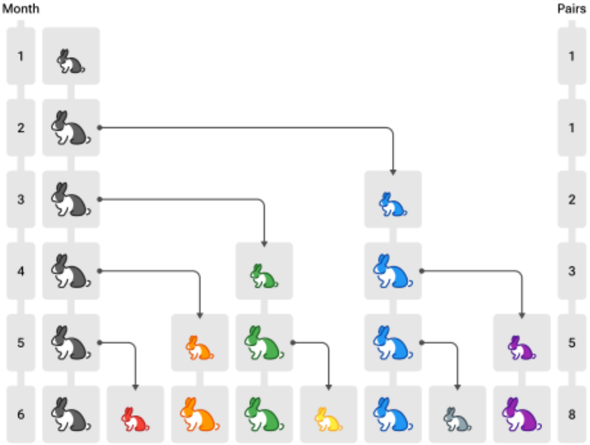
- Calculate extremas then output c:

$$c \mathrel{+}= \sum_{i=x-1}^{x+1} \mathbb{1}(\text{" } a_i \text{ is a extrema when } a_x = y \text{ "}) - \sum_{i=x-1}^{x+1} b_i$$

- if "P" query: $a_x = y$ and update b_{x-1}, b_x, b_{x+1}

Complexity: $\mathcal{O}(n + 3q)$

Problem 2: Fibonacci Sequence



Hình: Rabbit population

Fibonacci Sequence - Introduction Problem

The Fibonacci sequence is defined as:

$$
\begin{aligned}
F_0 &= 0 \\
F_1 &= 1 \\
F_n &= F_{n-1} + F_{n-2}, \quad \forall n > 1
\end{aligned}
$$

Problem: Given a positive integer n, calculate $F_n \mod (10^9 + 7)$.
Example:

- Input: 5
- Output: 5

Fibonacci Sequence - Approach

How many approach to solve this problem in descending order of complexity?

- Recursive $\mathcal{O}(2^n)$

Fibonacci Sequence - Approach

How many approach to solve this problem in descending order of complexity?

- Recursive $\mathcal{O}(2^n)$
- Dynamic Programming $\mathcal{O}(n)$

Fibonacci Sequence - Approach

How many approach to solve this problem in descending order of complexity?

- Recursive $\mathcal{O}(2^n)$
- Dynamic Programming $\mathcal{O}(n)$
- Matrix Exponential $\mathcal{O}(\log n)$

Fibonacci Sequence - Approach

How many approach to solve this problem in descending order of complexity?

- Recursive $\mathcal{O}(2^n)$
- Dynamic Programming $\mathcal{O}(n)$
- Matrix Exponential $\mathcal{O}(\log n)$
- Closed-form Expression $\mathcal{O}(\log n)$
  (Recall Closed-form Expression for everybody don't know or forgot)

$$F_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$$

Fibonacci Sequence - Approach

How many approach to solve this problem in descending order of complexity?

- Recursive $\mathcal{O}(2^n)$
- Dynamic Programming $\mathcal{O}(n)$
- Matrix Exponential $\mathcal{O}(\log n)$
- Closed-form Expression $\mathcal{O}(\log n)$
  (Recall Closed-form Expression for everybody don't know or forgot)

$$F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right]$$

- ...

Fibonacci Sequence - Approach

How many approach to solve this problem in descending order of complexity?

- Recursive $\mathcal{O}(2^n)$
- Dynamic Programming $\mathcal{O}(n)$
- Matrix Exponential $\mathcal{O}(\log n)$
- Closed-form Expression $\mathcal{O}(\log n)$
- ...

Due to **rounding error** (a phenomenon of digital computing resulting from the computer's inability to represent some numbers exactly) so using Closed-form Expression will be unpractical. So today we will learn about **Matrix Exponential** approach.

Fibonacci Sequence - Matrix Exponential

We can rewrite it to:

$$\begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} F_{n-1} \\ F_{n-2} \end{bmatrix}$$

So if we continue to rewrite the right part:

$$\begin{aligned} \begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix} &= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} F_{n-2} \\ F_{n-3} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} F_1 \\ F_0 \end{bmatrix} \end{aligned}$$

Then we get:

$$\begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{n-1} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

UIT
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Fibonacci Sequence - Fast Power

To more optimal we can use Divide and Conquer to calculate power of matrix

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n = \begin{cases} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, & \text{if } n = 1 \\ \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{n/2} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{n/2}, & \text{if } n \mod 2 = 0 \\ \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{\lfloor n/2 \rfloor} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{\lfloor n/2 \rfloor} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, & \text{if } n \mod 2 = 1 \end{cases}$$

Fibonacci Sequence - Moreover

At more general case:

- s_i stand for **start value** of F_i
- c_i stand for **coefficient** of F_i in the recurrence formula of F_n
- t stand for the number of variable in the recurrence formula of F_n

$$
\begin{aligned}
F_0 &= s_0 \\
F_1 &= s_1 \\
\dots \\
F_{t-1} &= s_{t-1} \\
F_n &= c_{n-1}F_{n-1} + c_{n-2}F_{n-2} + \dots + c_{n-t}F_{n-t}, \quad \forall n \geq t
\end{aligned}
$$

So how we can rewrite it in Matrix Exponential form?

Fibonacci Sequence - Moreover

Just like we do with Fibonacci's number we can rewrite it:

$$\begin{bmatrix} F_n \\ F_{n-1} \\ \cdots \\ F_{n-t+1} \end{bmatrix} = \begin{bmatrix} c_{n-1} & c_{n-2} & \cdots & c_{n-t} \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} F_{n-1} \\ F_{n-2} \\ \cdots \\ F_{n-t} \end{bmatrix}$$

$$= \begin{bmatrix} c_{n-1} & c_{n-2} & \cdots & c_{n-t} \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}^{n-t+1} \cdot \begin{bmatrix} s_0 \\ s_1 \\ \cdots \\ s_{n-t} \end{bmatrix}$$

Complexity: $\mathcal{O}(k^3 \log n)$

Problem 3: Twisted Treeline

Twisted Treeline

- Could someone summarize the statement?

Twisted Treeline

- Could someone summarize the statement?
- The problem is you need to find the square size $k \times k$ which has the most sum of value.

Twisted Treeline

- Could someone summarize the statement?
- The problem is you need to find the square size $k \times k$ which has the most sum of value.
- This is a classical DP problem in 2D space, but still exciting to follow.

Twisted Treeline

- Could someone summarize the statement?
- The problem is you need to find the square size $k \times k$ which has the most sum of value.
- This is a classical DP problem in 2D space, but still exciting to follow.
- We will discuss subtask-by-subtask of this problem.

Twisted Treeline

- Subtask 1: $1 \leq n, m \leq 100$, $k = 1$

Twisted Treeline

- Subtask 1: $1 \leq n, m \leq 100$, $k = 1$
- In this case, due to the size of the square is only 1, we only need to find the maximum value of whole matrix.

Twisted Treeline

- Subtask 1: $1 \leq n, m \leq 100$, $k = 1$
- In this case, due to the size of the square is only 1, we only need to find the maximum value of whole matrix.
- Time complextity: $\mathcal{O}(nm)$

Twisted Treeline

- Subtask 1: $1 \le n, m \le 100,\ k = 1$
- In this case, due to the size of the square is only 1, we only need to find the maximum value of whole matrix.
- Time complextity: $\mathcal{O}(nm)$
- Subtask 2: $1 \le n, m \le 100$

Twisted Treeline

- Subtask 1: $1 \leq n, m \leq 100$, $k = 1$
- In this case, due to the size of the square is only 1, we only need to find the maximum value of whole matrix.
- Time complextity: $\mathcal{O}(nm)$
- Subtask 2: $1 \leq n, m \leq 100$
- Think of a very naive algorithm. What should we do?

Twisted Treeline

- Subtask 1: $1 \leq n, m \leq 100$, $k = 1$
- In this case, due to the size of the square is only 1, we only need to find the maximum value of whole matrix.
- Time complextity: $\mathcal{O}(nm)$
- Subtask 2: $1 \leq n, m \leq 100$
- Think of a very naive algorithm. What should we do?
- 4-loop-in-a-row algorithm should work well, unless that your program has several bugs.

Twisted Treeline

- Subtask 1: $1 \leq n, m \leq 100,\ k = 1$
- In this case, due to the size of the square is only 1, we only need to find the maximum value of whole matrix.
- Time complextity: $\mathcal{O}(nm)$
- Subtask 2: $1 \leq n, m \leq 100$
- Think of a very naive algorithm. What should we do?
- 4-loop-in-a-row algorithm should work well, unless that your program has several bugs.
- You could choose every point in the matrix plays at a top-left point and the bottom-right point in the square, check whether it is a valid $k \times k$ square and update answer.

Twisted Treeline

- Subtask 1: $1 \leq n, m \leq 100,\ k = 1$
- In this case, due to the size of the square is only 1, we only need to find the maximum value of whole matrix.
- Time complextity: $\mathcal{O}(nm)$
- Subtask 2: $1 \leq n, m \leq 100$
- Think of a very naive algorithm. What should we do?
- 4-loop-in-a-row algorithm should work well, unless that your program has several bugs.
- You could choose every point in the matrix plays at a top-left point and the bottom-right point in the square, check whether it is a valid $k \times k$ square and update answer.
- Time complextity: $\mathcal{O}((nm)^2)$

Twisted Treeline

- Subtask 3: $1 \leq n, m \leq 10^3$, $k \leq 5$

Twisted Treeline

- Subtask 3: $1 \leq n, m \leq 10^3,\ k \leq 5$
- What is the most important constraint in this subtask?

Twisted Treeline

- Subtask 3: $1 \leq n, m \leq 10^3$, $k \leq 5$
- What is the most important constraint in this subtask?
- $k \leq 5$, very small constraint, we should focus on this.

Twisted Treeline

- Subtask 3: $1 \leq n, m \leq 10^3$, $k \leq 5$
- What is the most important constraint in this subtask?
- $k \leq 5$, very small constraint, we should focus on this.
- Assume that we fixed the top-left corner of the square, what is the time complexity of calculate the total value of the square?

Twisted Treeline

- Subtask 3: $1 \leq n, m \leq 10^3$, $k \leq 5$
- What is the most important constraint in this subtask?
- $k \leq 5$, very small constraint, we should focus on this.
- Assume that we fixed the top-left corner of the square, what is the time complexity of calculate the total value of the square?
- It's take $\mathcal{O}(k^2)$. Adding with the process of choosing top-left corner, it takes $\mathcal{O}(nmk^2)$.

- Subtask 3: $1 \leq n, m \leq 10^3$, $k \leq 5$
- What is the most important constraint in this subtask?
- $k \leq 5$, very small constraint, we should focus on this.
- Assume that we fixed the top-left corner of the square, what is the time complexity of calculate the total value of the square?
- It's take $\mathcal{O}(k^2)$. Adding with the process of choosing top-left corner, it takes $\mathcal{O}(nmk^2)$.

Twisted Treeline

- Subtask 4: $1 \leq n, m \leq 10^3$.

Twisted Treeline

- Subtask 4: $1 \leq n, m \leq 10^3$.
- Any ideas for this subtask?

Twisted Treeline

- Subtask 4: $1 \leq n, m \leq 10^3$.
- Any ideas for this subtask?
- Assume that we fixed the top-left corner of the square, how to reduce the time to calulate the total value?

Twisted Treeline

- Subtask 4: $1 \leq n, m \leq 10^3$.
- Any ideas for this subtask?
- Assume that we fixed the top-left corner of the square, how to reduce the time to calulate the total value?
- Assume that the area is $1 \times k$ instead of $k \times k$.

Twisted Treeline

- Subtask 4: $1 \leq n, m \leq 10^3$.
- Any ideas for this subtask?
- Assume that we fixed the top-left corner of the square, how to reduce the time to calulate the total value?
- Assume that the area is $1 \times k$ instead of $k \times k$.
- So the row is fixed, the problem become 1 dimension DP problem.

Twisted Treeline

- Subtask 4: $1 \leq n, m \leq 10^3$.
- Any ideas for this subtask?
- Assume that we fixed the top-left corner of the square, how to reduce the time to calulate the total value?
- Assume that the area is $1 \times k$ instead of $k \times k$.
- So the row is fixed, the problem become 1 dimension DP problem.
- $\text{sum}[i : i + k - 1] = \text{sum}[0 : i + k - 1] - \text{sum}[0 : i - 1]$ (3.1)

Twisted Treeline

- Subtask 4: $1 \leq n, m \leq 10^3$.
- Any ideas for this subtask?
- Assume that we fixed the top-left corner of the square, how to reduce the time to calulate the total value?
- Assume that the area is $1 \times k$ instead of $k \times k$.
- So the row is fixed, the problem become 1 dimension DP problem.
- $\text{sum}[i : i + k - 1] = \text{sum}[0 : i + k - 1] - \text{sum}[0 : i - 1]$ (3.1)
- Use a prefix sum to calculate $\text{sum}[i]$, hence (3.1) only takes $\mathcal{O}(1)$

Twisted Treeline

- We could upgrade it into 2 dimensional DP.

Twisted Treeline

- We could upgrade it into 2 dimensional DP.
- Let $S[i][j]$ be the sum of value of the rectangle with the top-left corner is $(1, 1)$ and bottom-right corner is (i, j).

Twisted Treeline

- We could upgrade it into 2 dimensional DP.
- Let $S[i][j]$ be the sum of value of the rectangle with the top-left corner is $(1, 1)$ and bottom-right corner is (i, j).
- What is the DP recursive formula?

Twisted Treeline

- We could upgrade it into 2 dimensional DP.
- Let $S[i][j]$ be the sum of value of the rectangle with the top-left corner is $(1,1)$ and bottom-right corner is (i,j).
- What is the DP recursive formula?
- $S[i][j] = S[i-1][j] + S[i][j-1] + a[i][j] - S[i-1][j-1]$

Twisted Treeline

- We could upgrade it into 2 dimensional DP.
- Let $S[i][j]$ be the sum of value of the rectangle with the top-left corner is $(1, 1)$ and bottom-right corner is (i, j).
- What is the DP recursive formula?
- $S[i][j] = S[i-1][j] + S[i][j-1] + a[i][j] - S[i-1][j-1]$
- How to calulate the sum of value with top-left conrner is (i, j) and bottom-right is $(i+k-1, j+k-1)$.

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

Twisted Treeline

- We could upgrade it into 2 dimensional DP.
- Let $S[i][j]$ be the sum of value of the rectangle with the top-left corner is $(1, 1)$ and bottom-right corner is (i, j).
- What is the DP recursive formula?
- $S[i][j] = S[i-1][j] + S[i][j-1] + a[i][j] - S[i-1][j-1]$
- How to calulate the sum of value with top-left conrner is (i, j) and bottom-right is $(i + k - 1, j + k - 1)$.
- It just takes $\mathcal{O}(1)$ sum $=$ $S[i+k-1][j+k-1] - S[i-1][j+k-1] - S[i+k-1][j-1] + S[i-1][j-1]$

Twisted Treeline

- We could upgrade it into 2 dimensional DP.
- Let $S[i][j]$ be the sum of value of the rectangle with the top-left corner is $(1, 1)$ and bottom-right corner is (i, j).
- What is the DP recursive formula?
- $S[i][j] = S[i-1][j] + S[i][j-1] + a[i][j] - S[i-1][j-1]$
- How to calulate the sum of value with top-left conrner is (i, j) and bottom-right is $(i+k-1, j+k-1)$.
- It just takes $\mathcal{O}(1)$ sum $=$ $S[i+k-1][j+k-1] - S[i-1][j+k-1] - S[i+k-1][j-1] + S[i-1][j-1]$
- The total complexity is $\mathcal{O}(nm)$.

- Subtask 5: $1 \leq n \times m \leq 2 \times 10^6$.

Twisted Treeline

- Subtask 5: $1 \leq n \times m \leq 2 \times 10^6$.
- This subtask is set to evaluate the programming skills of participants.

Twisted Treeline

- Subtask 5: $1 \leq n \times m \leq 2 \times 10^6$.
- This subtask is set to evaluate the programming skills of participants.
- Due to the size of input (and maybe output), you should put this code in the beginning of your main function:

```
ios_base::sync_with_stdio(false);
cin.tie(0); cout.tie(0);
```

Twisted Treeline

- Subtask 5: $1 \leq n \times m \leq 2 \times 10^6$.
- This subtask is set to evaluate the programming skills of participants.
- Due to the size of input (and maybe output), you should put this code in the beginning of your main function:
```
ios_base::sync_with_stdio(false);
cin.tie(0); cout.tie(0);
```
- Or you could use `scanf`, `printf`, `getchar`, `putchar`, ...

Problem 4: Massive mission

Massive mission

- Once again, could someone summarize the statement?

Massive mission

- Once again, could someone summarize the statement?
- In this problem we are given n rectangles, and we need to find the convex hull of them.

Massive mission

- Once again, could someone summarize the statement?
- In this problem we are given n rectangles, and we need to find the convex hull of them.
- It's clear that we only need to find the convex hull of $4n$ points.

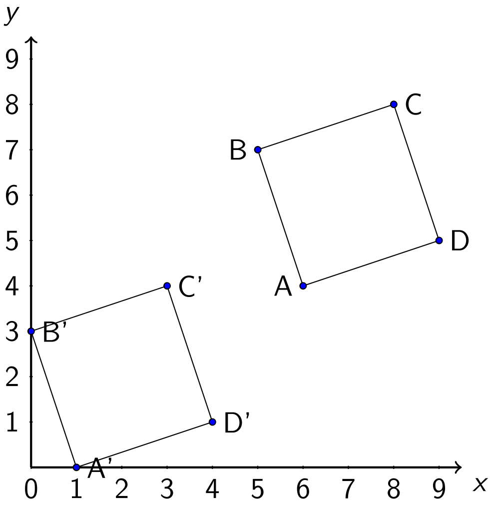
Massive mission

- Once again, could someone summarize the statement?
- In this problem we are given n rectangles, and we need to find the convex hull of them.
- It's clear that we only need to find the convex hull of $4n$ points.
- But what's the problem?

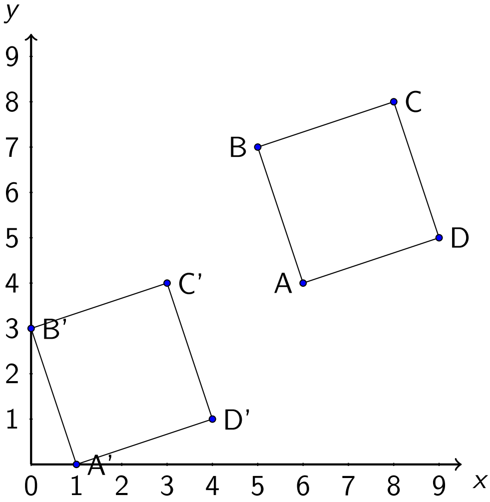
Massive mission

- Once again, could someone summarize the statement?
- In this problem we are given n rectangles, and we need to find the convex hull of them.
- It's clear that we only need to find the convex hull of $4n$ points.
- But what's the problem?
- We need to find 4 boundary points of a rectangle which the detail given from the problem.

Massive Mission

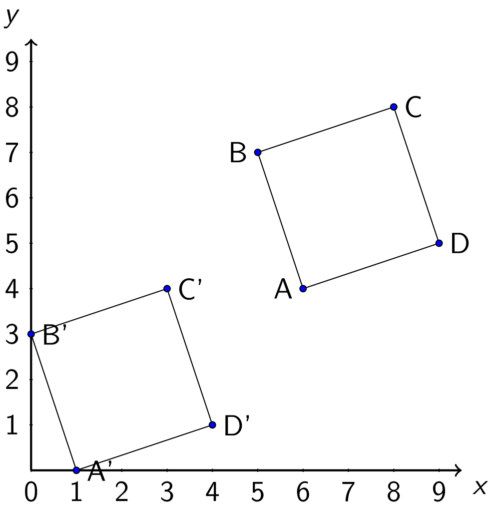


- It is very difficult if you want to find the rectangles directly from the information given by the problem.



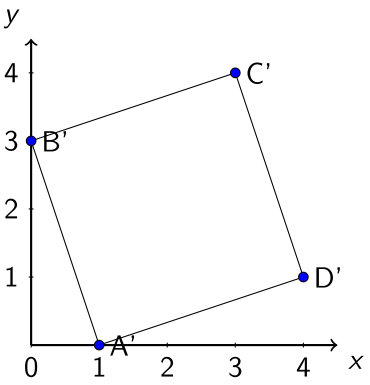
- It is very difficult if you want to find the rectangles directly from the information given by the problem.
- If we translate them to the original coordinate, it is clears to find the coordinates for 4 boundary points.

Massive Mission



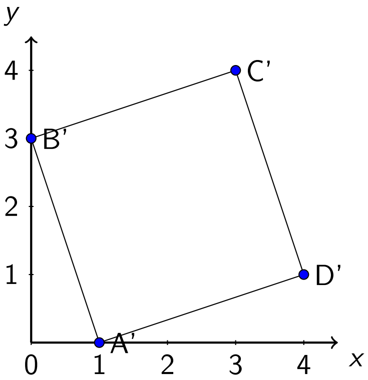
- It is very difficult if you want to find the rectangles directly from the information given by the problem.
- If we translate them to the original coordinate, it is clears to find the coordinates for 4 boundary points.
- Simple geometry formula could be used to find the A', B', C' and D', then use translation to find the original coordinates.

Massive Mission



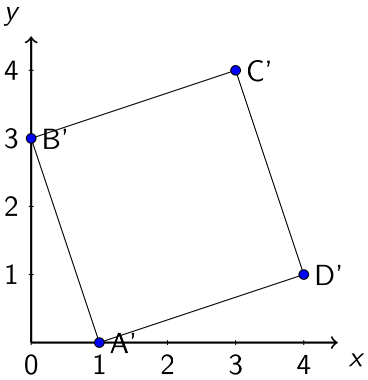
- We will try to find the coordinate of A' and B', from those, finding C' and D' would be clearly easy.

Massive Mission



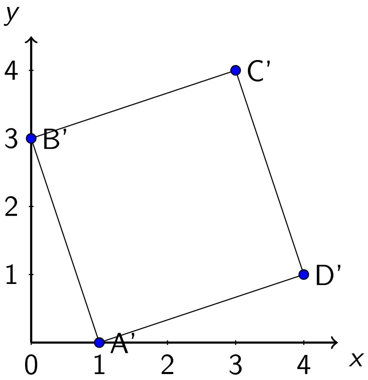
- We will try to find the coordinate of A' and B', from those, finding C' and D' would be clearly easy.
- We have $A'B' = h$ and $\widehat{A'B'O} = \phi$.

Massive Mission



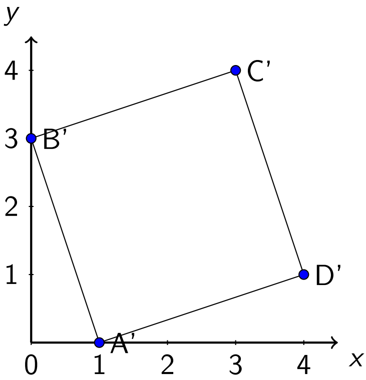
- We will try to find the coordinate of A' and B', from those, finding C' and D' would be clearly easy.
- We have $A'B' = h$ and $\widehat{A'B'O} = \phi$.
- We have $OB' = h\cos\phi$, so we got the coordinate of B'.

Massive Mission



- We will try to find the coordinate of A' and B', from those, finding C' and D' would be clearly easy.
- We have $A'B' = h$ and $\widehat{A'B'O} = \phi$.
- We have $OB' = h\cos\phi$, so we got the coordinate of B'.
- For the same method, we easily get the coordinate of A'.

Massive Mission



- We will try to find the coordinate of A' and B', from those, finding C' and D' would be clearly easy.
- We have $A'B' = h$ and $\widehat{A'B'O} = \phi$.
- We have $OB' = h\cos\phi$, so we got the coordinate of B'.
- For the same method, we easily get the coordinate of A'.
- Then C' and D'.

Massive Mission

- After that process, we will have a set of $4n$ points, then we should find the Convex Hull made from this set. Graham Algorithm or Mono chain technique (a.k.a Andrew Algorithm) are popular methods.

- After that process, we will have a set of $4n$ points, then we should find the Convex Hull made from this set. Graham Algorithm or Mono chain technique (a.k.a Andrew Algorithm) are popular methods.
- Let $S_i(x_i, y_i)$ be the n elements of the Convex Hull. Then its area is $S = \frac{1}{2}\sum_{i=1}^{n} |x_i y_{i+1} - x_{i+1} y_i|$. Let $S_{n+1} \equiv S_1$.

Massive Mission

- After that process, we will have a set of $4n$ points, then we should find the Convex Hull made from this set. Graham Algorithm or Mono chain technique (a.k.a Andrew Algorithm) are popular methods.
- Let $S_i(x_i, y_i)$ be the n elements of the Convex Hull. Then its area is $S = \frac{1}{2}\sum_{i=1}^{n} |x_i y_{i+1} - x_{i+1} y_i|$. Let $S_{n+1} \equiv S_1$.
- The final answer is $\frac{S_{rectangles}}{S}$.

Massive Mission

- After that process, we will have a set of $4n$ points, then we should find the Convex Hull made from this set. Graham Algorithm or Mono chain technique (a.k.a Andrew Algorithm) are popular methods.
- Let $S_i(x_i, y_i)$ be the n elements of the Convex Hull. Then its area is $S = \frac{1}{2}\sum_{i=1}^{n} |x_i y_{i+1} - x_{i+1} y_i|$. Let $S_{n+1} \equiv S_1$.
- The final answer is $\frac{S_{rectangles}}{S}$.
- Time complexity: $\mathcal{O}(n \log n)$, the complexity of creating the Convex Hull.

Conclusion

Conclusion

- During this session, we have discussed about:
  - Local-based approach in programming
  - Matrix Multiplication application in sequence calculation
  - 2 dimensional DP with inclusion-exclusion.
  - Geometry with translation and convex hull.
- Any questions?
- We hope you enjoy this session. Thank you.