Interna 2025/1 - O Poderoso Balão - Soluções

Enzo, Eric, Eduardo, Joãozão

Universidade do Estado de Santa Catarina
Créditos: Adaptado dos slides de IIT Delhi

7 de junho de 2025

Estatísticas

- 15 times presenciais.
- 44 times online.

N - Ni

Autor: Eric Grochowicz

- Imprima "Ni" K vezes.
- $N \leq 5$

Veredito	Quantidade
OK	15

N - Ni

Autor: Eric Grochowicz

- Imprima "Ni" K vezes.
- $N \leq 5$
- Complexidade: $\mathcal{O}(K)$.

E - Enigma

Autor: Eric Grochowicz

- Descobrir os valores A e B com:
  - $X = A + B$
  - $Y = |A - B|$
- $X, Y \leq 10^9$

Veredito	Quantidade
OK	14
WA	1

E - Enigma

Autor: Eric Grochowicz

- Sejam A, B tal que $A \geq B$
- $|A - B| = A - B$
- $A - B = Y$
- $A + B = X$

E - Enigma

Autor: Eric Grochowicz

- $2 \cdot A = X + Y$
- $B = X - A$
- Complexidade: $\mathcal{O}(1)$

M - Movieguessr

Autor: Eduardo Schwarz Moreira

- É Dado um array a de 1's e -1's (e o primeiro elemento é 0), um inteiro A e um inteiro B.
- Podemos definir um novo array $pref$ onde $pref_i$ é o somatório de a_j para $j \leq i$.

Veredito	Quantidade
WA	22
OK	13

M - Movieguessr

Autor: Eduardo Schwarz Moreira

- É Dado um array a de 1's e -1's (e o primeiro elemento é 0), um inteiro A (altura de Machado), um inteiro B (altura de Marcel) e a posição de Marcel na escada K.
- Podemos definir um novo array $pref$ onde $pref_i$ é o somatorio de a_j para $j \leq i$.
- A altura de Machado no i-ésimo degrau da escada é $pref_i + A$, e altura de Marcel é fixa e igual a $pref_K + B$.

M - Movieguessr

Autor: Eduardo Schwarz Moreira

- É Dado um array a de 1's e -1's (e o primeiro elemento é 0), um inteiro A (altura de Machado), um inteiro B (altura de Marcel) e a posição de Marcel na escada K.
- Podemos definir um novo array $pref$ onde $pref_i$ é o somatorio de a_j para $j \leq i$.
- A altura de Machado no i-ésimo degrau da escada é $pref_i + A$, e altura de Marcel é fixa e igual a $pref_K + B$.
- Para responder o problema, basta tentar todos os degraus (diferente de K), se $pref_i + A \geq pref_K + B$, i é uma resposta valida.
- Complexidade final: $O(N)$.

Problema B - Biblioteca Mágica

Autor: Eduardo Schwarz Moreira

- Dada uma string, retorne a string lexicograficamente mínima fazendo no máximo uma operação de swap entre dois índices.
- $N \leq 10^5$

Veredito	Quantidade
OK	12
WA	7
RT	1
TL	1

Problema B - Biblioteca Mágica

Autor: Eduardo Schwarz Moreira

- Solução: Compare a string original S com sort(S)
- Na primeira posição diferente i, faça o swap de S_i com S_j, sendo j a posição mais a direita da letra sort(S)$_i$ em S.

Problema B - Biblioteca Mágica

Autor: Eduardo Schwarz Moreira

- S = aaabbdda
- sort(S) = aaaabbdd

Problema B - Biblioteca Mágica

Autor: Eduardo Schwarz Moreira

- $S = \text{aaa}\mathbf{b}\text{bdda}$
- $\text{sort}(S) = \text{aaa}\mathbf{a}\text{bbdd}$
- $\mathbf{i = 3}$

Problema B - Biblioteca Mágica

Autor: Eduardo Schwarz Moreira

- $S = \text{aaa}\mathbf{b}\text{bdd}\color{red}{\text{a}}$
- $\text{sort}(S) = \text{aaa}\color{red}{\text{a}}\text{bbdd}$
- $\mathbf{i = 3}$
- $\color{red}{j = 7}$

Problema B - Biblioteca Mágica

Autor: Eduardo Schwarz Moreira

- $S = \text{aaa}\mathbf{b}\text{bdd}\color{red}{\text{a}}$
- $\text{sort}(S) = \text{aaa}\color{red}{\text{a}}\text{bbdd}$
- $\mathbf{i = 3}$
- $\color{red}{j = 7}$
- $\text{swap}(S_{\mathbf{3}}, S_{\color{red}{7}}) = \text{aaaabddb}$
- Complexidade: $\mathcal{O}(N \cdot \log N)$ (por causa do sort)

J - John Wick

Autor: Eduardo Schwarz Moreira

- Dado uma string com um # que se move da esquerda para a direita empurrando quaisquer o que ela encontre pela frente, simule o processo K vezes.
- $N, K \leq 500$

Veredito	Quantidade
OK	11
WA	1

J - John Wick

Autor: Eduardo Schwarz Moreira

- Dado uma string com um # que se move da esquerda para a direita empurrando quaisquer o que ela encontre pela frente, simule o processo K vezes.
- $N, K \leq 500$
- Para cada segundo, pode-se simular em $\mathcal{O}(N)$ um avanço.

J - John Wick

Autor: Eduardo Schwarz Moreira

- Dado uma string com um # que se move da esquerda para a direita empurrando quaisquer o que ela encontre pela frente, simule o processo K vezes.
- $N, K \leq 500$
- Para cada segundo, pode-se simular em $\mathcal{O}(N)$ um avanço.
- Complexidade: $\mathcal{O}(N \cdot K)$ (ou $\mathcal{O}(N)$, mas desnecessário)

H - Hierarquia Cinematográfica

Autor: Enzo de Almeida Rodrigues

- Dada uma árvore enraizada no vértice 1 com valores nos nodos, você pode escolher um conjunto disjunto de subárvores e somar seus valores. Informe a maior soma possível.
- $2 \leq N \leq 5 \cdot 10^5$

Veredito	Quantidade
OK	5
WA	1

- Solução: defina dp_u como a resposta para a subárvore do nodo u.

- Solução: defina dp_u como a resposta para a subárvore do nodo u.
- Transição da dp: podemos escolher entre pegar toda a subárvore de u ou não, portanto:
  - $dp_u = \max(\text{sum}_u, \sum_{v \in \text{Adj}_u} dp_v)$
  - sendo sum_u a soma dos valores dos nodos na subárvore de u
- A resposta do problema é dp_1

H - Hierarquia Cinematográfica

Autor: Enzo de Almeida Rodrigues

- Solução: defina dp_u como a resposta para a subárvore do nodo u.
- Transição da dp: podemos escolher entre pegar toda a subárvore de u ou não, portanto:
  - $dp_u = \max(\text{sum}_u, \sum_{v \in \text{Adj}_u} dp_v)$
  - sendo sum_u a soma dos valores dos nodos na subárvore de u
- A resposta do problema é dp_1
- Complexidade: $\mathcal{O}(N)$

- Bônus: o problema também pode ser resolvido em $\mathcal{O}(N \cdot \log N)$ de forma gulosa.

I - Idioma Perdido

Autor: Eric Grochowicz

- Dadas N strings, imprima uma ordenação do alfabeto de forma a tornar todas elas não-decrescentes lexicograficamente.
- $N \leq 10^5$
- $\sum |s_i| \leq 10^6$

Veredito	Quantidade
WA	11
OK	4

- Cada string s_i gera várias restrições do tipo $s_{i,j} \leq s_{i,j+1}$ para todo index j.

- Cada string s_i gera várias restrições do tipo $s_{i,j} \leq s_{i,j+1}$ para todo index j.
- Modelaremos o problema de forma a ter apenas restrições de **menor** em vez de menor ou igual:
  - Se $s_{i,j} = s_{i,j+1}$, ok
  - Senão, temos que $s_{i,j} < s_{i,j+1}$ tem que ser verdade.

I - Idioma Perdido

Autor: Eric Grochowicz

- Cada string s_i gera várias restrições do tipo $s_{i,j} \leq s_{i,j+1}$ para todo index j.
- Modelaremos o problema de forma a ter apenas restrições de **menor** em vez de menor ou igual:
  - Se $s_{i,j} = s_{i,j+1}$, ok
  - Senão, temos que $s_{i,j} < s_{i,j+1}$ tem que ser verdade.
- Podemos construir um grafo onde cada nodo é uma letra do alfabeto e as restrições são arestas entre os nodos.

I - Idioma Perdido

Autor: Eric Grochowicz

- Cada string s_i gera várias restrições do tipo $s_{i,j} \leq s_{i,j+1}$ para todo index j.
- Modelaremos o problema de forma a ter apenas restrições de **menor** em vez de menor ou igual:
  - Se $s_{i,j} = s_{i,j+1}$, ok
  - Senão, temos que $s_{i,j} < s_{i,j+1}$ tem que ser verdade.
- Podemos construir um grafo onde cada nodo é uma letra do alfabeto e as restrições são arestas entre os nodos.
- Para resolver o problema, precisamos apresentar uma ordem topológica dos nodos desse grafo. Se houver ciclos, não há resposta.
- Complexidade: $\mathcal{O}(\sum |s_i|)$.

F - Frodo & Sam

Autor: Eduardo Schwarz Moreira

- São dadas duas permutações P e Q de tamanho $N \leq 10^5$.
- Para cada subarray em comum de P e Q, se a é o começo do subarray em P e b o começo do subarray em Q, Frodo irá demorar $F * (a + b - 2)$ (onde a e b estão 1 indexado).

F - Frodo & Sam

Autor: Eduardo Schwarz Moreira

- São dadas duas permutações P e Q de tamanho $N \leq 10^5$.
- Para cada subarray em comum de P e Q, se a é o começo do subarray em P e b o começo do subarray em Q, Frodo irá demorar $F * (a + b - 2)$ tempo (onde a e b estão 1 indexado).
- Se c é o final do subarray em P e d o final do subarray em Q, Sam irá demorar $S * ((n - c) + (n - d))$ tempo (onde c e d estão 1 indexado).
- O tempo total para uma escolha de subarray é o máximo entre o tempo de Frodo e o tempo de Sam.

Veredito	Quantidade
OK	3

F - Frodo & Sam

Autor: Eduardo Schwarz Moreira

- Para cada subarray em comum de P e Q, se a é o começo do subarray em P e b o começo do subarray em Q, Frodo irá demorar $F * (a + b - 2)$ tempo (onde a e b estão 1 indexado).
- Se c é o final do subarray em P e d o final do subarray em Q, Sam irá demorar $S * ((n - c) + (n - d))$ tempo (onde c e d estão 1 indexado).
- O tempo total para uma escolha de subarray é o máximo entre o tempo de Frodo e o tempo de Sam.
- Podemos perceber que para um inicio de subarray i em P, O inicio do subarray em Q é determinado (basta achar a posição de P_i em Q).

F - Frodo & Sam

Autor: Eduardo Schwarz Moreira

- O tempo total para uma escolha de subarray é o máximo entre o tempo de Frodo e o tempo de Sam.
- Podemos perceber que para um inicio de subarray i em P, o inicio do subarray em Q é determinado (basta achar a posição de P_i em Q, denotarei essa posição como pos).
- Podemos aplicar um argumento guloso que, dado um i e pos_{P_i}, podemos avançar i e pos para a direita enquanto $P_{i+1} = Q_{pos+1}$

F - Frodo & Sam

Autor: Eduardo Schwarz Moreira

- Podemos aplicar um argumento guloso que, dado um i e pos_{P_i}, podemos avançar i e pos para a direita enquanto $P_{i+1} = Q_{pos+1}$
- Quando não conseguirmos mais avançar, podemos calcular a resposta do subarray atual, e começar o próximo subarray da posição que paramos.
- Complexidade total: $O(N)$.
- OBS: outros algoritimos de Longest Commom Substring também funcionariam, como usar Hash ou estruturas de dados mais avançadas.

K - Kill Bill Vol. 1

Autor: Enzo de Almeida Rodrigues

- É dado um array de tamanho N, sendo cada elemento de uma cor.
- Operação: escolher duas posições (i, j),sendo $i < j$ e $c_i = c_j$ e atribuir $c_k = c_i$ para todo $i \leq k \leq j$.
- $N \leq 2 \cdot 10^5$

Veredito	Quantidade
WA	5
OK	2
TL	1

- dp_i: menor número de cores diferentes possíveis para o prefixo terminado na posição i.

- dp_i: menor número de cores diferentes possíveis para o prefixo terminado na posição i.
- $dp_i = \min(dp_{i-1} + 1, \min dp(c_i))$
  - sendo $\min dp(c_i)$ o menor valor entre todos os dp_j tal que $c_j = c_i$

- dp_i: menor número de cores diferentes possíveis para o prefixo terminado na posição i.
- $dp_i = \min(dp_{i-1} + 1, \min dp(c_i))$
  - sendo $\min dp(c_i)$ o menor valor entre todos os dp_j tal que $c_j = c_i$
- Intuição: a resposta ótima com certeza não possui dois segmentos disjuntos de mesma cor.
- Complexidade: $O(N \log N)$.

Problema D - Dentro da Matriz

Autor: Eric Grochowicz

- Dada uma matriz $N \times M$, informe o menor OR bitwise de um caminho saindo de $(1, 1)$ e de (X, Y) até (N, M).
- $N \cdot M \leq 10^5$

Veredito	Quantidade
WA	1
OK	1

Problema D - Dentro da Matriz

Autor: Eric Grochowicz

- Algoritmo de Dijkstra não funciona.
  - O OR sempre aumenta permanece igual, parecido com a soma quando só há valores positivos.
  - Porém, Dijkstra parte do fato de que um menor caminho tradicional sempre pode ser obtido através da extensão de outros menores caminhos.

Problema D - Dentro da Matriz

Autor: Eric Grochowicz

- Solução: Conseguimos construir a resposta passando pelos bits do maior para o menor.

Problema D - Dentro da Matriz

Autor: Eric Grochowicz

- Solução: Conseguimos construir a resposta passando pelos bits do maior para o menor.
- Façamos uma função que vê o menor caminho de uma célula qualquer (I, J) até (N, M). Rodaremos essa função partindo de $(1, 1)$ e de (X, Y).
  - Estando no k-ésimo bit, e com a resposta construída para os bits mais significativos que k sendo ans_{k+1}, vamos tentar ligar tal bit:

Problema D - Dentro da Matriz

Autor: Eric Grochowicz

- Solução: Conseguimos construir a resposta passando pelos bits do maior para o menor.
- Façamos uma função que vê o menor caminho de uma célula qualquer (I, J) até (N, M). Rodaremos essa função partindo de $(1, 1)$ e de (X, Y)
- Estando no k-ésimo bit, e com a resposta construída para os bits mais significativos que k sendo ans_{k+1}, vamos tentar ligar tal bit:
  - Se conseguirmos ir de (I, J) para (N, M) com a restrição de só podermos passar por $a_{i,j}$ se $a_{i,j} \leq ans_{k+1} + 2^k$; podemos acender o bit 2^k e dizer que $ans_k = ans_{k+1} + 2^k$.

Problema D - Dentro da Matriz

Autor: Eric Grochowicz

- Solução: Conseguimos construir a resposta passando pelos bits do maior para o menor.
- Façamos uma função que vê o menor caminho de uma célula qualquer (I, J) até (N, M). Rodaremos essa função partindo de $(1, 1)$ e de (X, Y)
- Estando no k-ésimo bit, e com a resposta construída para os bits mais significativos que k sendo ans_{k+1}, vamos tentar ligar tal bit:
  - Caso contrário, fazemos $ans_k = ans_{k+1}$ e continuamos o processo.
  - Complexidade final: $\mathcal{O}(N \cdot M \cdot \log(\text{MAX}))$.

O - Marciano

Autor: Enzo de Almeida Rodrigues

- Dado um grafo de N nodos e M arestas, eu tenho D escadas e D destinos que quero alcançar. Com uma escada com valor e_i, eu consigo passar por qualquer aresta cujo valor é menor ou igual a e_i. Sabendo que posso usar qualquer escada para ir a qualquer destino e, uma vez que uso uma escada para um destino, eu a perco, diga qual o máximo de destinos é possível alcançar.
- $N, M \leq 10^5$
- $D \leq N$

Veredito	Quantidade
OK	1

- Se eu tenho uma escada com valor e_i, eu quero usar ela em um caminho cuja maior aresta é menor ou igual a e_i.

- Se eu tenho uma escada com valor e_i, eu quero usar ela em um caminho cuja maior aresta é menor ou igual a e_i.
- É ótimo escolher os caminhos de forma a **minimizar** a **maior** aresta de cada um deles.

- Se eu tenho uma escada com valor e_i, eu quero usar ela em um caminho cuja maior aresta é menor ou igual a e_i.
- É ótimo escolher os caminhos de forma a minimizar a maior aresta de cada um deles.
  - Computaremos a árvore geradora mínima (MST) para encontrar a maior aresta no caminho da origem a cada um dos D destinos.

- Se eu tenho uma escada com valor e_i, eu quero usar ela em um caminho cuja maior aresta é menor ou igual a e_i.
- É ótimo escolher os caminhos de tal forma a minimizar a menor aresta de cada um deles.
  - Computaremos a árvore geradora mínima (MST) para encontrar a maior aresta no caminho da origem a cada um dos D destinos.
- Podemos parear a maior escada com o maior caminho para o qual ela pode servir e assim por diante de forma gulosa.
- Complexidade: $\mathcal{O}(M \cdot \log M + D \cdot \log D)$

Problema A - Absolute Cinema

Autor: João Marcos de Oliveira

- Dado um array de tamanho N, informe o somatório do produto do máximo e do mínimo para todo segmento contínuo.
- $1 \leq N \leq 10^5$

Problema A - Absolute Cinema

Autor: João Marcos de Oliveira

- Técnica: Divisão e Conquista

Problema A - Absolute Cinema

Autor: João Marcos de Oliveira

- Técnica: Divisão e Conquista
  - `solve(l, r) = calcula(l, r, mid) + solve(l, mid) + solve(mid + 1, r)`
  - A resposta do problema é `solve(0, n - 1)`

Problema A - Absolute Cinema

Autor: João Marcos de Oliveira

- Técnica: Divisão e Conquista
  - `solve(l, r) = calcula(l, r, mid) + solve(l, mid) + solve(mid + 1, r)`
  - A resposta do problema é `solve(0, n - 1)`
  - Para implementar a função `calcula`, utilizamos de prefixos/sufixos de máximo, de mínimo e de soma.

Problema A - Absolute Cinema

Autor: João Marcos de Oliveira

- Técnica: Divisão e Conquista
  - `solve(l, r) = calcula(l, r, mid) + solve(l, mid) + solve(mid + 1, r)`
  - A resposta do problema é `solve(0, n - 1)`
  - Para implementar a função `calcula`, utilizamos de prefixos/sufixos de máximo, de mínimo e de soma.
- Complexidade: $\mathcal{O}(N \cdot \log N)$

Problema A - Absolute Cinema

Autor: João Marcos de Oliveira

- Técnica: Divisão e Conquista
  - `solve(l, r) = calcula(l, r, mid) + solve(l, mid) + solve(mid + 1, r)`
  - A resposta do problema é `solve(0, n - 1)`
  - Para implementar a função `calcula`, utilizamos de prefixos/sufixos de máximo, de mínimo e de soma.
- Complexidade: $\mathcal{O}(N \cdot \log N)$
- Existem soluções alternativas em $O(N \cdot \log^2 N)$, utilizando de Segment Tree.

Problema C - Clichês

Autor: João Marcos de Oliveira

- São dados vários padrões e uma string S, informe qual a maior quantidade aparições de padrões (pode repetir) sendo que pode-se escolher um range de S para inverter.
- $|S| \leq 2000$

Problema C - Clichês

Autor: João Marcos de Oliveira

- Sem a parte de inverter, esse é um problema conhecido que podemos resolver utilizando o autômato de Aho-Corasick.

Problema C - Clichês

Autor: João Marcos de Oliveira

- Sem a parte de inverter, esse é um problema conhecido que podemos resolver utilizando o autômato de Aho-Corasick.
- Com a inversão, a mesma solução levaria tempo $O(N^3)$ para contar as ocorrências de padrões testando todas as possibilidades de inversão.

Problema C - Clichês

Autor: João Marcos de Oliveira

- Sem a parte de inverter, esse é um problema conhecido que podemos resolver utilizando o autômato de Aho-Corasick.
- Com a inversão, a mesma solução levaria tempo $O(N^3)$ para contar as ocorrências de padrões testando todas as possibilidades de inversão.
- A sacada é que podemos pré computar o avanço no autômato, usando binary lifting.

Problema C - Clichês

Autor: João Marcos de Oliveira

- Sem a parte de inverter, esse é um problema conhecido que podemos resolver utilizando o autômato de Aho-Corasick.
- Com a inversão, a mesma solução levaria tempo $O(N^3)$ para contar as ocorrências de padrões testando todas as possibilidades de inversão.
- A sacada é que podemos pré computar o avanço no autômato, usando binary lifting.
  - $v = go(i, u, 2^k)$
  - $go(i, u, 2^{k+1}) = go(i + 2^k, v, 2^k)$

Problema C - Clichês

Autor: João Marcos de Oliveira

- Sem a parte de inverter, esse é um problema conhecido que podemos resolver utilizando o autômato de Aho-Corasick.
- Com a inversão, a mesma solução levaria tempo $O(N^3)$ para contar as ocorrências de padrões testando todas as possibilidades de inversão.
- A sacada é que podemos pré computar o avanço no autômato, usando binary lifting.
  - $v = go(i, u, 2^k)$
  - $go(i, u, 2^{k+1}) = go(i + 2^k, v, 2^k)$
- Complexidade: $\mathcal{O}(N^2 \cdot \log N)$

G - Godfather

Autor: João Marcos de Oliveira

- Eric e João jogam em turnos, em cada turno, o jogador atual tem dois valores X e Y. Os possíveis movimentos em um turno são:
  - Transformar (X, Y) em $(X + Y, X)$
  - Transformar (X, Y) em $(X + Y, Y)$

G - Godfather

Autor: João Marcos de Oliveira

- Eric e João jogam em turnos, em cada turno, o jogador atual tem dois valores X e Y. Os possíveis movimentos em um turno são:
  - Transformar (X, Y) em $(X + Y, X)$
  - Transformar (X, Y) em $(X + Y, Y)$
- Pode-se enxergar que essas operações são transformações lineares, portanto podem ser aplicadas em sequência com multiplicação de matrizes.

G - Godfather

Autor: João Marcos de Oliveira

- Eric e João jogam em turnos, em cada turno, o jogador atual tem dois valores X e Y. Os possíveis movimentos em um turno são:
  - Transformar (X, Y) em $(X + Y, X)$
  - Transformar (X, Y) em $(X + Y, Y)$
- Pode-se enxergar que essas operações são transformações lineares, portanto podem ser aplicadas em sequência com multiplicação de matrizes.
- Para responder cada query de inversão de turnos em um intervalo (l, r), podemos usar uma Segment Tree Lazy que guarda o produto das matrizes em cada intervalo.

G - Godfather

Autor: João Marcos de Oliveira

- Eric e João jogam em turnos, em cada turno, o jogador atual tem dois valores X e Y. Os possíveis movimentos em um turno são:
  - Transformar (X, Y) em $(X + Y, X)$
  - Transformar (X, Y) em $(X + Y, Y)$
- Pode-se enxergar que essas operações são transformações lineares, portanto podem ser aplicadas em sequência com multiplicação de matrizes.
- Para responder cada query de inversão de turnos em um intervalo (l, r), podemos usar uma Segment Tree Lazy que guarda o produto das matrizes em cada intervalo.
- Complexidade: $\mathcal{O}((N + Q) \cdot \log N \cdot 2^3)$

L - Las Tortuguitas

Autor: Eduardo Schwarz Moreira

- É dado (implicitamente) uma matrix r onde a i-ésima linha é a soma de prefixo da linha anterior, e a primeira linha é uma PA com termo inicial X e razão x com $N \leq 3 \cdot 10^5$ linhas e $K = 3 \cdot 10^5$ colunas.
- O problema pede $Q \leq 3 \cdot 10^5$ perguntas de soma em range dessa matrix.

L - Las Tortuguitas

Autor: Eduardo Schwarz Moreira

- É dado (implicitamente) uma matrix r onde a i-ésima linha é a soma de prefixo da linha anterior, e a primeira linha é uma PA com termo inicial X e razão x com $N \leq 3 \cdot 10^5$ linhas e $K = 3 \cdot 10^5$ colunas.
- O problema pede $Q \leq 3 \cdot 10^5$ perguntas de soma em range dessa matrix.
- Se mudarmos um pouco a definição, podemos perceber que $r_{i,j} = r_{i-1,j} + r_{i,j-1}$
- Com essa definição, podemos perceber que r_i tem os K primeiros elementos da $i + 1$ coluna de pascal, (onde todos os elementos estão multiplicados pela constante X).

L - Las Tortuguitas

Autor: Eduardo Schwarz Moreira

- É dado (implicitamente) uma matrix r onde a i-ésima linha é a soma de prefixo da linha anterior, e a primeira linha é uma PA com termo inicial X e razão x com $N \leq 3 \cdot 10^5$ linhas e $K = 3 \cdot 10^5$ colunas.
- O problema pede $Q \leq 3 \cdot 10^5$ perguntas de soma em range dessa matrix.
- Se mudarmos um pouco a definição, podemos perceber que $r_{i,j} = r_{i-1,j} + r_{i,j-1}$
- Com essa definição, podemos perceber que r_i tem os K primeiros elementos da $i+1$ coluna de pascal, (onde todos os elementos estão multiplicados pela constante X).
- Para calcular a resposta de uma query i, L, R, podemos calcular $r_{i+1,R} - r_{i+1,L-1}$, que podem ser calculados rapidamente com binômio de Newton.
- Complexidade final: $O(N + K)$.

Obrigado!