Aula 3

Estruturas de Prefixo

Maratona de Programação

FEI

May 10, 2025

Fundamento: queries estáticas de soma

Problema: dado um vetor A de tamanho N ($N \leq 10^6, |A_i| \leq 10^9$) , responda Q queries ($Q \leq 10^6$) do tipo:

Dados dois índices l e r ($1 \leq l < r \leq N$),
computar a soma dos elementos do vetor A de l até r.
Formalmente, respoder para cada query:

$$\sum_{i=l}^{r} A_i = A_l + A_{l+1} + \cdots + A_r$$

Fundamento: queries estáticas de soma

Exemplos

Fundamento: queries estáticas de soma

Exemplos

$$l = 1, r = 5$$

A :

7	5	2	3	2	7	8	5	1
1	2	3	4	5	6	7	8	9

Fundamento: queries estáticas de soma

Exemplos

$$l = 1, r = 5$$

A :

7	**5**	**2**	**3**	**2**	7	8	5	1
1	2	3	4	5	6	7	8	9

$$l = 5, r = 8$$

A :

7	5	2	3	**2**	**7**	**8**	**5**	1
1	2	3	4	5	6	7	8	9

Fundamento: queries estáticas de soma

Exemplos

$l = 1, r = 5$

A :

7	**5**	**2**	**3**	**2**	7	8	5	1
1	2	3	4	5	6	7	8	9

$l = 5, r = 8$

A :

7	5	2	3	**2**	**7**	**8**	**5**	1
1	2	3	4	5	6	7	8	9

$l = 4, r = 6$

A :

7	5	2	**3**	**2**	**7**	8	5	1
1	2	3	4	5	6	7	8	9

Fundamento: queries estáticas de soma

Ideia de solução 1: brutar cada query

Fundamento: queries estáticas de soma

Ideia de solução 1: brutar cada query

- Ideia: Para cada query, iterar de l até r no vetor e ir acumulando A_i.

Fundamento: queries estáticas de soma

Ideia de solução 1: brutar cada query

- Ideia: Para cada query, iterar de l até r no vetor e ir acumulando A_i.
- Isso com certeza da o resultado certo, mas ...

Fundamento: queries estáticas de soma

Ideia de solução 1: brutar cada query

- Ideia: Para cada query, iterar de l até r no vetor e ir acumulando A_i.
- Isso com certeza da o resultado certo, mas ...
- Por que essa solução **não** é ótima?

Fundamento: queries estáticas de soma

Ideia de solução 1: brutar cada query

- Ideia: Para cada query, iterar de l até r no vetor e ir acumulando A_i.
- Isso com certeza da o resultado certo, mas ...
- Por que essa solução **não** é ótima?
- Complexidade de tempo: $\mathcal{O}(N * Q)$.

Fundamento: queries estáticas de soma

Ideia de solução 1: brutar cada query

- Ideia: Para cada query, iterar de l até r no vetor e ir acumulando A_i.
- Isso com certeza da o resultado certo, mas ...
- Por que essa solução **não** é ótima?
- Complexidade de tempo: $\mathcal{O}(N * Q)$.
- Quantidade de operacoes no pior caso: $10^6 * 10^6 = 10^{12}$.

Fundamento: queries estáticas de soma

Ideia de solução 1: brutar cada query

- Ideia: Para cada query, iterar de l até r no vetor e ir acumulando A_i.
- Isso com certeza da o resultado certo, mas ...
- Por que essa solução **não** é ótima?
- Complexidade de tempo: $\mathcal{O}(N * Q)$.
- Quantidade de operacoes no pior caso: $10^6 * 10^6 = 10^{12}$.
- Resultado: **Time Limit Exceeded (TLE)**.

Fundamento: queries estáticas de soma

Ideia de solução 2: pré-computar todas as possíveis queries

Fundamento: queries estáticas de soma

Ideia de solução 2: pré-computar todas as possíveis queries

- Ideia: Antes das queries, criar um map ou um vetor bidimensional que guarda a resposta pra query (l, r).

Fundamento: queries estáticas de soma

Ideia de solução 2: pré-computar todas as possíveis queries

- Ideia: Antes das queries, criar um map ou um vetor bidimensional que guarda a resposta pra query (l, r).
- preencher esse vetor com a resposta de cada uma dessa queries.

Fundamento: queries estáticas de soma

Ideia de solução 2: pré-computar todas as possíveis queries

- Ideia: Antes das queries, criar um map ou um vetor bidimensional que guarda a resposta pra query (l, r).
- preencher esse vetor com a resposta de cada uma dessa queries.
- Por que essa solução **não** é ótima?

Fundamento: queries estáticas de soma

Ideia de solução 2: pré-computar todas as possíveis queries

- Ideia: Antes das queries, criar um map ou um vetor bidimensional que guarda a resposta pra query (l, r).
- preencher esse vetor com a resposta de cada uma dessa queries.
- Por que essa solução **não** é ótima?
- Quantas queries **distintas** podem ter?

Fundamento: queries estáticas de soma

Ideia de solução 2: pré-computar todas as possíveis queries

- Ideia: Antes das queries, criar um map ou um vetor bidimensional que guarda a resposta pra query (l, r).
- preencher esse vetor com a resposta de cada uma dessa queries.
- Por que essa solução **não** é ótima?
- Quantas queries **distintas** podem ter?
  - Melhor perguntando: quantos pares ordenados (l, r), com l e r $(1 \leq l < r \leq N)$ existem?

Fundamento: queries estáticas de soma

Ideia de solução 2: pré-computar todas as possíveis queries

- Ideia: Antes das queries, criar um map ou um vetor bidimensional que guarda a resposta pra query (l, r).
- preencher esse vetor com a resposta de cada uma dessa queries.
- Por que essa solução **não** é ótima?
- Quantas queries **distintas** podem ter?
  - Melhor perguntando: quantos pares ordenados (l, r), com l e r $(1 \leq l < r \leq N)$ existem?
  - $\frac{N*(N-1)}{2}$ (uma quantidade quadrática).

Fundamento: queries estáticas de soma

Ideia de solução 2: pré-computar todas as possíveis queries

- Ideia: Antes das queries, criar um map ou um vetor bidimensional que guarda a resposta pra query (l, r).
- preencher esse vetor com a resposta de cada uma dessa queries.
- Por que essa solução **não** é ótima?
- Quantas queries **distintas** podem ter?
  - Melhor perguntando: quantos pares ordenados (l, r), com l e r $(1 \leq l < r \leq N)$ existem?
  - $\frac{N*(N-1)}{2}$ (uma quantidade quadrática).
- Complexidade de tempo e espaço: $\mathcal{O}(N^2)$.

Fundamento: queries estáticas de soma

Ideia de solução 2: pré-computar todas as possíveis queries

- Ideia: Antes das queries, criar um map ou um vetor bidimensional que guarda a resposta pra query (l, r).
- preencher esse vetor com a resposta de cada uma dessa queries.
- Por que essa solução **não** é ótima?
- Quantas queries **distintas** podem ter?
  - Melhor perguntando: quantos pares ordenados (l, r), com l e r $(1 \leq l < r \leq N)$ existem?
  - $\frac{N*(N-1)}{2}$ (uma quantidade quadrática).
- Complexidade de tempo e espaço: $\mathcal{O}(N^2)$.
- Quantidade de operacoes no pior caso: $10^6 * 10^6 = 10^{12}$.

Fundamento: queries estáticas de soma

Ideia de solução 2: pré-computar todas as possíveis queries

- Ideia: Antes das queries, criar um map ou um vetor bidimensional que guarda a resposta pra query (l, r).
- preencher esse vetor com a resposta de cada uma dessa queries.
- Por que essa solução **não** é ótima?
- Quantas queries **distintas** podem ter?
  - Melhor perguntando: quantos pares ordenados (l, r), com l e r $(1 \leq l < r \leq N)$ existem?
  - $\frac{N*(N-1)}{2}$ (uma quantidade quadrática).
- Complexidade de tempo e espaço: $\mathcal{O}(N^2)$.
- Quantidade de operacoes no pior caso: $10^6 * 10^6 = 10^{12}$.
- Resultado: **Time Limit Exceeded (TLE) ou MLE**.

Fundamento: queries estáticas de soma

Ideia de solução 3: Soma de Prefixo

Fundamento: queries estáticas de soma

Ideia de solução 3: Soma de Prefixo

- Ideia: Para cada query, buscamos o primeiro i tal que $A_i \geq X$.

Fundamento: queries estáticas de soma

Ideia de solução 3: Soma de Prefixo

- Ideia: Para cada query, buscamos o primeiro i tal que $A_i \geq X$.
- Começamos considerando todo o segmento $[1, N]$ como possível resposta.

Fundamento: queries estáticas de soma

Ideia de solução 3: Soma de Prefixo

- Ideia: Para cada query, buscamos o primeiro i tal que $A_i \geq X$.
- Começamos considerando todo o segmento $[1, N]$ como possível resposta.
- Escolhemos o valor no meio do segmento e verificamos se ele é $\geq X$.

Fundamento: queries estáticas de soma

Ideia de solução 3: Soma de Prefixo

- Ideia: Para cada query, buscamos o primeiro i tal que $A_i \geq X$.
- Começamos considerando todo o segmento $[1, N]$ como possível resposta.
- Escolhemos o valor no meio do segmento e verificamos se ele é $\geq X$.
- Se for, ele pode ser uma resposta, mas ainda tentamos encontrar uma posição melhor à esquerda, olhando agora para $[1, \mathit{meio}]$.

Fundamento: queries estáticas de soma

Ideia de solução 3: Soma de Prefixo

- Ideia: Para cada query, buscamos o primeiro i tal que $A_i \geq X$.
- Começamos considerando todo o segmento $[1, N]$ como possível resposta.
- Escolhemos o valor no meio do segmento e verificamos se ele é $\geq X$.
- Se for, ele pode ser uma resposta, mas ainda tentamos encontrar uma posição melhor à esquerda, olhando agora para $[1, meio]$.
- Se não for, descartamos ele e toda a parte à esquerda, olhando agora para $(meio, N]$.

Fundamento: queries estáticas de soma

Ideia de solução 3: Soma de Prefixo

- Ideia: Para cada query, buscamos o primeiro i tal que $A_i \geq X$.
- Começamos considerando todo o segmento $[1, N]$ como possível resposta.
- Escolhemos o valor no meio do segmento e verificamos se ele é $\geq X$.
- Se for, ele pode ser uma resposta, mas ainda tentamos encontrar uma posição melhor à esquerda, olhando agora para $[1, meio]$.
- Se não for, descartamos ele e toda a parte à esquerda, olhando agora para $(meio, N]$.
- Repetimos até o sobrar só um valor. A posição final encontrada (se houver) é a menor com valor $\geq X$.