

Maratona de Programação

Semana de Informática 2026

André G. Santos Salles V. G. Magalhães

Departamento de Informática
Universidade Federal de Viçosa (UFV), Brasil

Semana de Informática, 2026

B. Busca binária (29/29)¹

Resumo

- Ano de nascimento de três irmãs no time
- Logue busca idade de Binária, a irmã do meio

Solução

- Encontrar o valor do meio e diminuir de 2026

¹quantos times dos 29 participantes passaram a questão e número de submissões

D. De volta para o centenário (29/37)

Resumo

- Estudantes começam uma viagem no tempo em 2126
- Em cada viagem, adicionam X anos ao ano atual
- Após cada uma, estão antes ou depois do centenário da UFV, 2026?
- Ex.: $-50, -60, -90, 100$ passariam pelos anos 2076, 2016, 1926, 2026.

Solução

- Simular as viagens e verificar se o destino é antes ou depois de 2026

F. Fatia do bolo (28/36)

Resumo

- Bolo de tamanho $H \times V$
 - fatiado em F_1 pedaços iguais na horizontal
 - um deles fatiado em F_2 pedaços iguais na vertical
 - e assim por diante, alternadamente
- Encontrar a área da última fatia

Solução

- Dividir H por F_1 , dividir V por F_2 , dividir o resultado H/F_1 por F_3 , ...
- A área final é, então, $\frac{H}{F_1 \times F_3 \times F_5 \times \dots} \times \frac{V}{F_2 \times F_4 \times F_6 \times \dots}$
- Note que isso é o mesmo que $\frac{H \times V}{F_1 \times F_2 \times F_3 \times F_4 \times F_5 \times F_6 \times \dots}$
- Não precisa alternar, basta dividir pelo produto dos cortes

Obs.: melhor dividir por F_i sucessivamente que dividir pelo produto deles, para evitar overflow (para facilitar, os limites da entrada evitavam overflow)

C. Cupom Fiscal (26/32)

Resumo

- Agrupar ocorrências de um mesmo produto em um cupom fiscal de supermercado.
- Exemplo:

Entrada	Saída
1 arroz 3 feijao 4 arroz	5 arroz 3 feijao

Solução $\mathcal{O}(N)$ ou $\mathcal{O}(N \log N)$

- Usar um dicionário para guardar os produtos e quantidades.
- Como temos que imprimir na ordem da primeira ocorrência, passar duas vezes pela entrada: uma contando ocorrências e outra imprimindo o total e marcando os já impressos

Solução alternativa $\mathcal{O}(N^2)$

- Usar um array para guardar os produtos e quantidades
 - Para cada produto, percorrer o array até encontrá-lo e somar a quantidade
 - Se não for encontrado, acrescentá-lo no final com a quantidade comprada
 - Por fim, imprimir o array em ordem
- *Obs.: a eficiência poderia ser um problema se a entrada fosse maior.*

I. Idade quadrado perfeito (26/45)

Resumo

- Quem nasceu em 1980 completou 45 anos em $2025 = 45^2$
- Quem nasceu em 1892 completou 44 anos em $1926 = 44^2$
- Informar se nascidos no ano A fazem parte do grupo quadrado perfeito
 - Ou seja, se completam x anos no ano x^2 , para algum x

Solução matemática: $\mathcal{O}(1)$

- Nascidos no ano A completam x anos no ano $A + x$
- Verificar se há x com $A + x = x^2$
- Reescrevendo, temos $x^2 - x - A = 0$
- verificar se a solução $x = \frac{1 \pm \sqrt{1+4A}}{2}$ é inteira

Solução força bruta: $\mathcal{O}(N)$

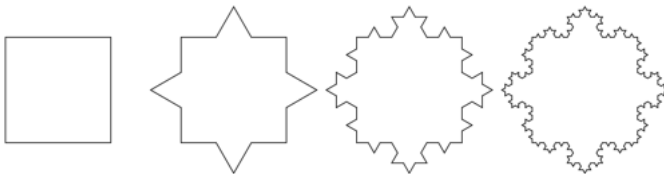
- São poucos anos, pode testar todos os valores
- Começando de $x = 1$, verificar se $A + x = x^2$; parar quando $x^2 > A + x$

Resumo

- Curva de Koch níveis 0, 1, 2, 3:



- A cada nível, um segmento se torna 4 segmentos de $1/3$ do tamanho
- Calcular área e perímetro da piscina cujos lados são N níveis de Koch:



- Fractal: um objeto geométrico complexo cujas partes menores se parecem com o todo, repetindo o mesmo padrão em diferentes escalas



Solução matemática:

- O comprimento da curva é uma PG de razão $4/3$
 - Nível 0, lado de tamanho L . Nível 1, 4 segmentos de $1/3$ de L , total $\frac{4}{3}L$. Cada um gera 4 segmentos de $1/3$ de $\frac{4}{3}L$ no nível 2, total $\frac{4 \times 4}{3 \times 3}L$.
 - De forma geral, total $\frac{4^n}{3^n}L$ no nível n . Contando os 4 lados: $P = 4 \times \frac{4^n}{3^n}L$
- Cada nível acrescenta triângulos na área, PG's de razão 4 e de razão $1/3$
 - No nível 1, um triângulo de lado $L/3$ acrescenta $\frac{(\frac{L}{3})^2 \sqrt{3}}{4}$ na área². No nível 2, 4 triângulos de lado $L/9$ acrescentam $4 \times \frac{(\frac{L}{9})^2 \sqrt{3}}{4}$, ...
 - Contando o quadrado e os 4 lados: $A = L^2 + 4 \times \sum_{i=1}^N 4^{i-1} \times \frac{(\frac{L}{3^i})^2 \sqrt{3}}{4}$

²Um triângulo equilátero de lado ℓ tem área $\frac{\ell^2 \sqrt{3}}{4}$.

- Curva de Koch: 

Solução recursiva:

- Fractal é naturalmente recursivo
- $\text{CompLado}(n) = \frac{4}{3} \times \text{CompLado}(n-1)$, com $\text{CompLado}(0) = L$
- $\text{Área}(L, n) = \Delta(L/3) + 4 \times \text{Área}(L/3, n-1)$, com $\text{Área}(L, 0) = 0$
(*área do triângulo central mais área nos 4 segmentos com um nível a menos*)
- $P = 4 \times \text{comprimento}$ e $A = 4 \times \text{área acrescida mais o quadrado central}$.

E. Expandindo a expressão (14/30)

Resumo

- Expandir expressão $(a + b + c + \dots)^2$
- Exemplo, se $N = 3$: $(a + b + c)^2 = a^2 + b^2 + c^2 + 2ab + 2ac + 2bc$

Solução 1 (mais genérica) $\mathcal{O}(N^2 \log N)$

- Gerar todos os termos contando quantas vezes aparece (usar um dicionário)
- Ordenar os termos e imprimir a expressão

Solução 2 $\mathcal{O}(N^2)$

- Gerar todos os termos já na ordem

Para i de 0 a $N - 1$

Para j de 0 a $N - 1$

Se $i == j$ escrever L_i^2

Senão escrever $2L_iL_j$

Dica: considerar que as variáveis são inteiros de 0 a $N - 1$. Converter para letras ($0 \rightarrow a, 27 \rightarrow a1$) apenas na hora de imprimir a saída. No código acima, L_i é a letra $i \% 26 + 'a'$ seguida de $i / 26$ se $i > 26$

G. Geração de Armstrong (9/27)

Resumo

- Gerar números de Armstrong no intervalo $[a, b]$
- Um número N de d dígitos é um número de Armstrong
 $\iff N = \text{soma de cada dígito de } N \text{ elevado a } d$
- Ex.: $153 = 1^3 + 5^3 + 3^3$

Solução $\mathcal{O}(N \log N)$ – TLE

- Testar cada número do intervalo verificando se é um número de Armstrong
- Custo $\mathcal{O}(N \log N)$, muito lento porque a entrada pode ser até 2×10^8

Solução $\mathcal{O}(1)$

- $\mathcal{O}(N \log N)$ não é tão ineficiente, é possível processar o pior caso em $< 10s$
- Há poucos números de Armstrong (menos de 30 para a pior entrada)
→ gerar lista de números localmente (pré-processar) e colocá-los no código

A. Alterando a senha do Sapiens (5/10)

Resumo

- Trocar senha A em B fazendo o mínimo de operações
- Senhas podem ser apenas algumas palavras válidas (entrada)
- Sistema aceita trocar uma senha por outra apenas se elas diferirem em 1 letra
- Exceção: podem ser feitas trocas “especiais” se a diferença for 2, mas isso pode ser feito no máximo 2 vezes.

Solução

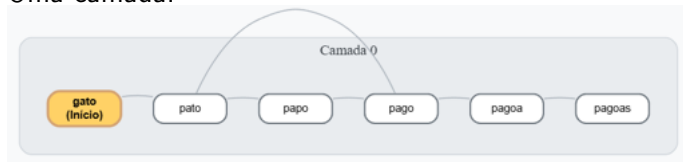
- Se não houvesse trocas especiais: bastaria criar um grafo onde os vértices são palavras e existe uma aresta entre duas senhas sse é possível trocar uma pela outra.
- Então, achar caminho mínimo (BFS).

A. Alterando a senha do Sapiens 📌 (5/10)

Solução $\mathcal{O}(N^2)$

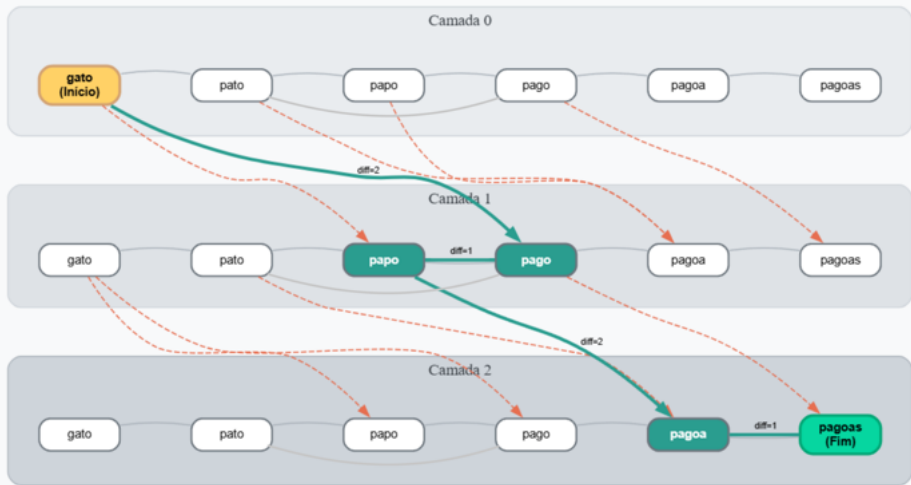
- Gerar grafo com 3 camadas (cópias do grafo), onde a camada 0 indica que não foi usada nenhuma troca especial, a 1 indica que foi usada uma troca especial, ...
- Se uma palavra X pode ser transformada em Y com uma troca especial, colocar uma aresta direcionada de X em cada camada para Y da camada seguinte.
- Aresta direcionada: uma vez que a pessoa usou uma troca especial, ela “gasta” essa troca $\rightarrow$ camadas limitam qtd de trocas especiais.

Uma camada:



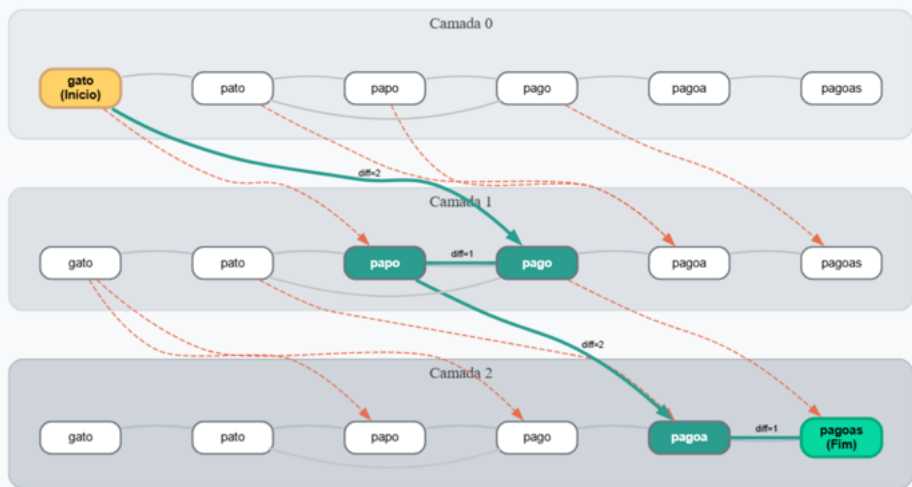
A. Alterando a senha do Sapiens 📌 (5/10)

As 3 camadas. Solução: caminho mínimo de gato até pagoas em alguma das 3 camadas.



A. Alterando a senha do Sapiens (5/10)

As 3 camadas. O caminho destacado abaixo é apenas para ilustrar as trocas – o mínimo é gato → pago → pagoas.



J. Jogo do morango 🍓 (5/33)

Resumo

- Jogadores se alternam cortando o bolo; perde quem ficar com o morango



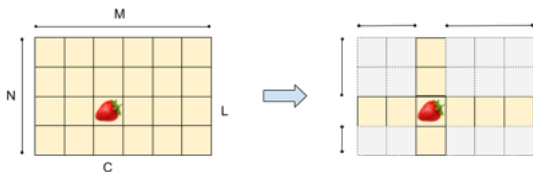
Solução PD: $\mathcal{O}(N^2 M^2)$

- Criar função $solve(n, m, l, c)$, que retorna *true* sse quem estiver fazendo a jogada atual conseguir fazer com que o outro fique com o morango (perca).
- Sua função deverá testar todas $n - 1 + m - 1$ possibilidades de corte e ver se em alguma delas você consegue forçar o outro a perder.
- Para cada possibilidade, você “corta o bolo” e vê se a chamada recursiva no pedaço com morango (jogada do adversário) retorna *false* (adversário não conseguiu ganhar).
- Se você conseguir ganhar em alguma (fazer o outro perder) $\rightarrow$ return *true*.
- Usar matriz 4D $(N \times M \times N \times M)$ para memoizar valores de $solve(n, m, l, c)$.

J. Jogo do morango 🍓 (5/33)

Solução NIM: $\mathcal{O}(1)$

- O jogo consiste em cortar fora linhas e colunas ao redor do morango



- Em cada jogada, o jogador pode retirar quanto quiser em dos 4 lados



- Jogo de Nim com pilhas de tamanho $C-1, L-1, N-L, M-C$
- Se o XOR dos valores for zero, quem começa vence; senão, perde.

H. Hoje é dia de pesar (2/24)

Resumo

- Há uma fórmula “complicada” $A(i)$, que calcula o peso de uma capivara i , para $i = 0 \dots N - 1$.
- Achar um i QUALQUER cujo peso seja maior ou igual ao dos vizinhos (máximo local).
- Ex.: $A(i) = 1, 7, 9, 6, 0, 3, 1 \rightarrow 2$ ou 5 seriam respostas válidas.

Solução Força bruta: $\mathcal{O}(QN)$ – TLE

- Para cada valor de 1 a $N - 2$, testar se $A_i \geq A_{i-1}$ e $A_i \geq A_{i+1}$
- São até $Q = 10^5$ consultas com até $N = 10^8$ capivaras, muito lento

H. Hoje é dia de pesar (2/24)

Solução Busca binária: $\mathcal{O}(Q \log N)$

- Se $A(\text{meio}) \geq \text{os dois vizinhos} \rightarrow$ return meio
- Se $A(\text{meio}) < A(\text{meio} + 1) \rightarrow$ certamente existe um máximo na direita (pode ser até o b , caso a função apenas cresça) $\rightarrow$ chame recursivamente de $\text{meio} + 1$ até b
- O mesmo vale para o outro lado
- Note que se $A(\text{meio}) > A(\text{meio} - 1) \rightarrow$ pode não haver máximo em $[a, \text{meio} - 1]$ (figura de baixo)
- O mesmo para o outro lado.

