

Relatório 1º projeto ASA 2025/2026

Grupo: AL065

Aluno(s): Gonçalo Rafael (113528) e Nuno Teixeira (114559)

Descrição do Problema e da Solução

A solução apresentada trata-se de um caso de programação dinâmica com recurso a tabulação (bottom-up), onde se optou por descobrir o último aminoácido a ser removido para encontrar a energia máxima, visto que começando pelo primeiro criaria dependências que causariam um problema na resolução, pois os subproblemas $[l, k-1]$ e $[k+1, r]$ teriam o aminoácido em $k-1$ e $k+1$ como vizinhos. Por isso, resolveu-se escolher o último aminoácido a ser removido em cada um dos subproblemas, tornando-os independentes.

Começa-se por receber os dados do input onde a sequência de aminoácidos é convertida para o índice correspondente na tabela de afinidades de aminoácidos. De seguida, criam-se duas matrizes, *optimal* e *cache*, de forma a guardar tanto a energia máxima para um determinado intervalo, tal como o índice do aminoácido ótimo. Definem-se as variáveis temporárias *total*, *maxEnergia* e *maxK* de modo a tornar o código mais eficiente (o que evita estar constantemente a encontrar o endereço nas matrizes *optimal* e *cache*), entra-se, por fim, no *loop*. Dentro dos *fors* atribui-se às respetivas tabelas os valores de energia ótima de remoção e do índice do aminoácido ótimo a remover. Ao completar as tabelas, faz-se um último passo para conseguir a sequência ótima de remoção, utilizando uma recursão. De modo a evitar problemas com números muito grandes (e sabendo que os números não serão negativos) utiliza-se unsigned long long para que não ocorra nenhum problema de overflow.

Análise Teórica

Leitura de dados: $O(n)$

Leitura input potenciais	$O(n)$
Leitura input sequência + conversão:	$O(n)$

Processamento das instâncias: $O(n^2)$

Init cache[n+2][n+2]	$O(n^2)$
Init optimal[n+2][n+2]	$O(n^2)$

Algoritmo: $O(n^3)$

For l = n to 1	$O(n)$
For r = l to n	$O(n)$
maxEnergia = 0	$O(1)$
maxK = 1	$O(1)$
For k = l to r	$O(n)$
total = Custo Remoção	$O(1)$
if(total > maxEnergia)	$O(1)$
maxEnergia = total	$O(1)$
maxK = k	$O(1)$
if(total == maxEnergia && k >= maxK)	$O(1)$
maxK = K	$O(1)$
cache[l][r] = maxEnergia	$O(1)$
optimal[l][r] = k	$O(1)$

Relatório 1º projeto ASA 2025/2026

Grupo: AL065

Aluno(s): Gonçalo Rafael (113528) e Nuno Teixeira (114559)

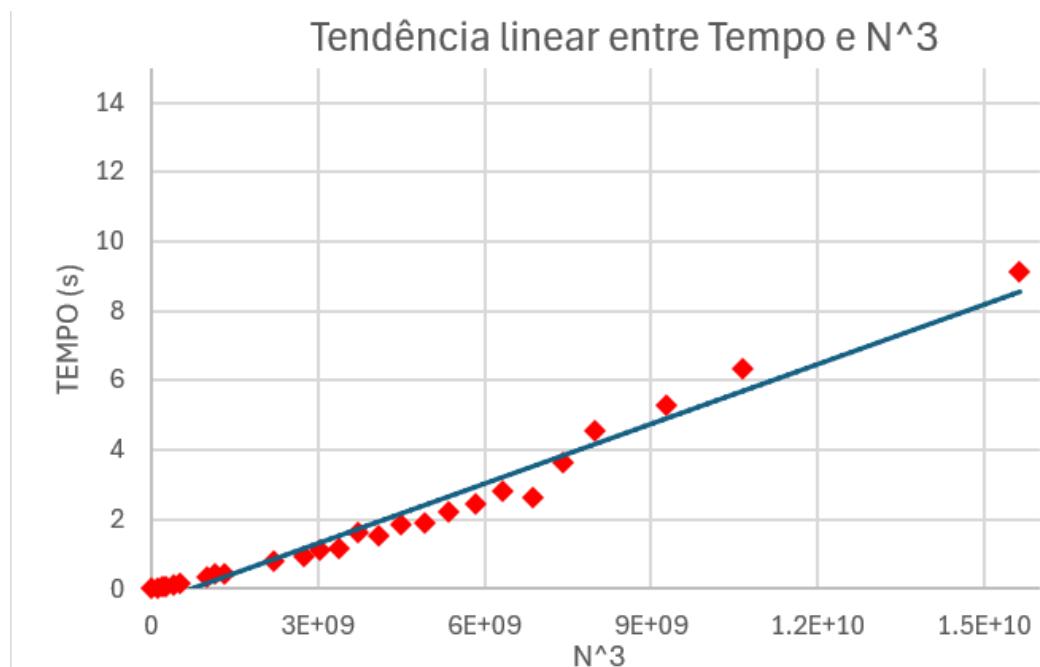
Output: $O(n)$

printEnergia	$O(1)$
Computar Sequência Ótima	$O(n)$

Complexidade global: $O(n^3)$

Avaliação Experimental dos Resultados

Foram geradas 26 instâncias de tamanho incremental, de $n=100$ até $n=2500$. Observáveis no gráfico, onde o eixo dos XX representa o n^3 , e o eixo dos YY o tempo.



É possível observar uma relação linear, que confirma a complexidade global $O(n^3)$.