Relatório do 1.º Projeto ASA 2025/2026

Grupo: AL090

Alunos: Tiago Andrês (ist1113875) e Rodrigo Baptista (ist1114851)

Descrição do Problema e da Solução:

É-nos dada uma cadeia de n aminoácidos, cada um com uma classe (P, N, A, B) e um potencial. Ao remover um aminoácido, a energia libertada depende do seu potencial, dos potenciais dos vizinhos e das afinidades entre as classes. O objetivo é encontrar a ordem de remoção que maximiza a energia total.

A solução proposta recorre a programação dinâmica em intervalos. Definimos dp[l][r] como a energia máxima obtida ao remover todos os aminoácidos entre l e r. Para cada intervalo e para cada possível primeira remoção k, combinamos as soluções ótimas dos subintervalos (l,k) e (k,r) com a energia libertada por remover k. Preenchendo os intervalos por ordem crescente de tamanho, obtemos dp e a tabela split, que regista a melhor escolha de k. No final, dp[0][n+1] contém a energia máxima, e a ordem ótima é reconstruída recursivamente a partir de split.

Análise Teórica:

Pseudo código:

```
// Ler input

Ler n, pot, classes

// DP

Para todos os intervalos [l,r]:

    Para k em (l,r):

        Dp[l][r] = melhor opção usando k

// Reconstrução

build(0, n+1) usando split

Imprimir resultado
```

Análise:

A leitura do input (n, potenciais e classes) é linear, O(n), sendo adicionadas duas sentinelas em 0 e n+1. A fase principal usa programação dinâmica em intervalos: existem O(n^2) intervalos [l,r] e, para cada um, testam-se O(n) possíveis primeiras remoções k, resultando numa complexidade O(n^3). A solução final é

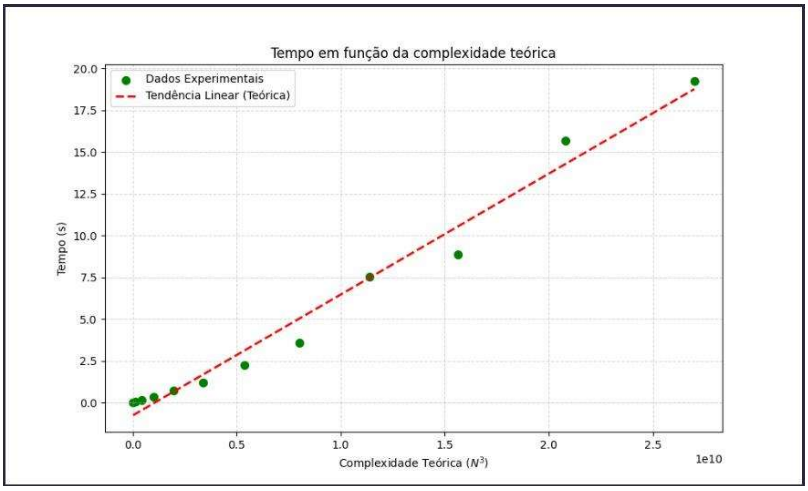
Relatório do 1.º Projeto ASA 2025/2026

Grupo: AL090

Alunos: Tiago Andrês (ist1113875) e Rodrigo Baptista (ist1114851)

reconstruída a partir da tabela split, num processo linear O(n), tal como a impressão da saída. Assim, a complexidade total é dominada pela DP, $O(n^3)$. Em memória, as tabelas dp e split têm dimensão (n+2)×(n+2), o que implica um consumo $O(n^2)$.

Avaliação Experimental dos Resultados:



N	N^3	Tempo (s)
250	1,56E+07	0,007
500	1,25E+08	0,061
750	4,22E+08	0,135
1000	1,00E+09	0,332
1250	1,95E+09	0,702
1500	3,38E+09	1,176
1750	5,36E+09	2,236
2000	8,00E+09	3,578
2250	1,14E+10	7,517
2500	1,56E+10	8,868
2750	2,08E+10	15,689
3000	2,70E+10	19,244

Observando o gráfico, verifica-se que os tempos de execução crescem de forma muito próxima de uma função cúbica, como previsto teoricamente. A representação dos tempos experimentais em função de n^3 produz praticamente uma linha reta, o que indica que o fator dominante do algoritmo é de facto a componente de programação dinâmica com custo $O(n^3)$. Além disso, os desvios em relação à tendência são reduzidos, confirmando que a implementação é estável e que os custos adicionais são negligenciáveis face ao custo cúbico principal. Assim, os resultados experimentais validam claramente a análise teórica realizada.