Thiago Lutz Dias RGA: 2021.1906.048-4

Lista 3

O texto comenta sobre como uma maneira de encontrar padrões no sequenciamento do DNA foi resolvido. O desafio era identificar todas as strings de tamanho $2|s|$ que são possíveis substrings de uma string s, dado o set de todas as substrings de tamanho k de S. Era necessário um algoritmo que, após concatenar todas as strings, verificasse se todos os resultados eram permitidos seguindo a lógica previamente descrita.

A partir desse contexto, eles tentaram utilizar uma árvore binária de busca. Entretanto, o algoritmo ainda era muito lento. Portanto, após análise, verificaram que a operação mais comum no algoritmo era a busca, e alteraram a estrutura de dados para uma tabela hash. A tabela hash tinha uma performance quase que o dobro em questão de velocidade que a árvore previamente usada, mas ainda não era o suficiente.

Após mais análise, decidiram tentar utilizando uma árvore de sufixos. Dessa vez, a velocidade foi boa, mas a quantidade de memória usada era muito grande e o programa acabava falhando. Foi, então, realizado um ajuste e utilizaram uma árvore de sufixos comprimida. Ela resolveu o problema com sucesso.

A partir dessa história pode-se ver que, em alguns casos, a estrutura dos dados pode ser o fator decisivo para um algoritmo estar correto e solucionar o desafio, ou falhar e não auxiliar na resolução do problema. Essa história também nos mostra que até mesmo ótimos programadores erram. É normal precisar refazer/melhorar o que já foi feito. Por fim, essa história nos mostra uma boa maneira de lidar com problemas difíceis: iniciar com uma solução simples e, a partir dela, desenvolver soluções melhores, e não necessariamente mais frequentemente, mais complexas.