

Sprint 4

Pedro Cunial

May 26, 2017

Contents

1 Roteiro e Estimativa de Tempo	1
--	----------

1 Roteiro e Estimativa de Tempo

Pretendo fazer uma aula mais lúdica, de forma que a necessidade de uso do algoritmo viria de uma "história".

A "história" seria que o governo japonês está buscando pelo "Escolhido", que seria o filho do Pai Mei. Por saberem que o "Escolhido" se encontra no Brasil dando aula no Insper, o governo japonês acabou me contratando para que descobrisse quem de fato é este "Escolhido".

Inicialmente, minha sugestão é que ele fosse o professor Marcelo Hashimoto, que capturassemos ele e o entregássemos ao governo nipônico. No entanto, lembrando que sua matéria ainda não havia terminado e ainda tinha pendências acadêmicas com o mesmo decidi pesquisar melhor sobre como descobrir com uma maior certeza se o professor de fato é o "Escolhido". Pretendo gastar no máximo 3 minutos nesta introdução, preferencialmente me contendo a 2 minutos.

Com o contexto introduzido, a ideia é já sugerir a análise de DNA como o melhor método para o caso e, como estudante de computação, minha ideia seria implementar um código que fizesse a mesma. Sob este contexto, explicarei a análise de subsequências tal como o próprio conceito de subsequência contra o de substring. Para esta parte da apresentação não pretendo passar de 1 minuto (totalizando 4 minutos no pior caso até agora).

Daí vem uma primeira implementação, onde, em pseudo-código, na lousa escreverei um código bastante ingênuo e que não se baseia em programação dinâmica para resolver o problema. Apesar da minha ideia inicial ser a de fazer um código em Python, acredito que o setup do telão e erros tolos

da minha parte podem comprometer muito da aula, não valendo a maior "utilidade" do produzido. Enfim, a ideia é narrar conforme escrevo esta implementação, algo como "Para uma letra 'i' da minha sequência A, a letra 'j' da sequência B é igual a ela? Se sim (...)". Isso não deve se estender por muito tempo, levando no máximo 3 minutos, mas também não muito menos (totalizando 7 minutos até agora no pior caso).

Com ela, a ideia é mostrar sua desnecessária complexidade e redundância ao simular a execução do código em uma espécie de teste de mesa, mostrando os cálculos redundantes nas chamadas recursivas ao fazer o desenho de uma árvore das chamadas e circulando os valores já calculados e chamados novamente, o que não deve levar mais de 1 minuto (totalizando 8 minutos até agora).

Com isso, a ideia é sugerir o uso da memoização para a solução deste problema, no entanto, a estrutura a ser utilizada não é clara, assim como a forma que faremos ela. Ai entra o material auxiliar, onde entregarei a grupos de 4 ou mais alunos um material muito semelhante ao utilizado na aula onde você ensinou pilhas, filas, arrays e matrizes, ou seja, um tubo fechado aberto de só um lado, um tubo fechado aberto dos dois lados, um tabuleiro quadriculado e uma sequência de quadrados. A ideia é que os alunos decidam a melhor estrutura para armazenar os valores já calculados dos respectivos 'i's e 'j's das sequências (sendo "i" e "j" índices de seus caracteres, conceito que eles já devem estar familiarizados pela parte anterior da aula). Para isso, não gostaria de passar de 3 minutos pensando entre eles (pois a ideia é que os grupos rapidamente reconheçam que a matriz é a melhor ideia entre as opções) e mais um minuto (talvez um pouco menos até) para que apresentassem sua conclusão aos outros, totalizando 12 minutos até agora.

Com a discussão da melhor estrutura, é esperado que pelo menos um dos grupos tenha chegado em uma forma razoável de como utilizar a matriz para armazenar estes dados, tal que posso gastar os últimos minutos adaptando o pseudo-código feito com a nova implementação sugerida.

Para fechar a aula, a ideia era dizer que após aplicar o algoritmo, acabei descobrindo que o "Escolhido" era na verdade o professor Luciano Soares, filho renegado de Pai Mei.