

INF 213 - Roteiro da Aula Prática

Arquivos disponibilizados:

<https://drive.google.com/file/d/1HCxSINZLTLq44WMV2FWxgr3yi14eJXUb/view?usp=sharing>

Etapa 1

a) Considere o programa “adivinharComplexidade.cpp”. Compile-o (sem usar a flag -O3) e teste-o usando a sintaxe “./a.out N” (onde N é o tamanho da entrada).

Qual a complexidade do programa “adivinharComplexidade.cpp” ? Faça testes com $N=5,6,7,8,\dots,13$ e tente adivinhar a complexidade dele (não tente entendê-lo ou olhar na internet o que a função `next_permutation` faz).

A complexidade é $O(n!)$

b) Considerando o programa “adivinharComplexidade2.cpp”, compile-o (sem usar a flag -O3) e teste-o usando a sintaxe “./a.out N” (onde N é o tamanho da entrada).

Teste o programa com vários valores de N e analise o código.

Por que as funções “dfjkhbjknbjkcjfhui” e “dfjkhbjknbjkcjfhui2”, apesar de muito parecidas se comportam de forma tão diferente em relação ao tempo de execução?

Qual complexidade a função “find” (da STL do C++) parece ter? (descubra isso apenas medindo os tempos de execução, não tente entendê-la, pesquisar na internet, perguntar ao Chat GPT, etc).

A complexidade do find é $O(n)$, que combinado com a complexidade do for resulta em uma complexidade $O(n^2)$

Qual complexidade a função “log” (do C++) parece ter? Descubra isso apenas medindo os tempos de execução -- dica: teste com números muito maiores (exemplo: 500 milhões, 1 bilhão, 2 bilhões) e apague a chamada à segunda função para conseguir fazer essa medição apenas da primeira (caso contrário não dará tempo do programa terminar antes do deadline desta prática, que é ainda neste século).

Aparentemente a complexidade do log é $O(1)$

Etapa 2

Faça a análise de complexidade das funções presentes no arquivo `analise1.cpp` (tais funções podem nem compilar ou fazer sentido -- estamos interessados apenas na complexidade dos algoritmos).

Escreva suas respostas como comentários no topo das respectivas funções (veja o exemplo na primeira função de `analise1.cpp`). Lembre-se de sempre usar a notação “O”, simplificar ao máximo a resposta final (ou seja, em vez de $O(3n^4 + n^3)$ a resposta deverá ser algo como $O(n^4)$) e utilizar a complexidade mais justa (conforme visto em aula).

Considere sempre o pior caso de cada função. (a não ser que dito o contrário nos comentários) Preste bastante atenção a todas funções!

Lembrem-se sempre de pedir ajuda ao professor se necessário (não fiquem em dúvida sobre a complexidade de alguma função).

Submissao da aula pratica:

A solucao deve ser submetida utilizando o sistema submittty (submittty.dpi.ufv.br). Envie `analise1.cpp` pelo submittty. Envie também um PDF deste documento após terminar as respostas da Etapa 1 (o nome do arquivo deverá ser `roteiro.pdf`).