

PROJETO 1, 2, 3

Análise empírica de algoritmos de busca, ordenação e inserção/remoção em sequência

- O que é?

Análise empírica é o processo de avaliação do funcionamento de um algoritmo através da observação experimental do seu comportamento em diferentes instâncias do problema proposto.

Neste projeto, vocês deverão realizar uma análise empírica de complexidade de tempo dos algoritmos de busca, ordenação e inserção/remoção em TAD sequência (vetor e lista ligada) apresentados em sala de aula: busca sequencial e busca binária; ordenação por inserção, por seleção, quicksort e mergesort; inserção/remoção no início, no fim ou em posição aleatória utilizando vetor ou lista ligada. Esta análise deverá demonstrar com dados experimentais qual é a complexidade de tempo de melhor caso, pior caso e caso médio destes algoritmos em função do tamanho do vetor.

- Mas, como?

Primeiramente vocês irão implementar os algoritmos de busca **binária** e **sequencial**; ordenação **por inserção**, **por seleção**, **quicksort** e **mergesort**; **inserção** e **remoção** em uma **lista ligada** e em um **vetor**; a forma pode ser **iterativa** ou **recursiva**, fica ao critério do aluno.

Em seguida, vocês deverão encontrar um método de medir quanto tempo o algoritmo demora para se realizar as operações. As duas alternativas são:

- (1) medir o tempo utilizando um código de medição;
- (2) incluir no código um contador de passos.

Assim, os passos para gerar uma observação são:

- (1) gerar uma instância do problema de ordenação: um vetor de tamanho N;
- (2) medir o tempo (ou número de passos) requerido pelo algoritmo para ordenar o vetor.

Para permitir a análise estatística deverão ser realizadas múltiplas observações com o mesmo valor paramétrico, isto é, o tamanho do vetor (N). Quanto maior o número de amostras para o mesmo parâmetro, maior confiança você terá de que a estatística computada reflete o comportamento do algoritmo em análise.

Por fim, os dados estatísticos obtidos da observação do tempo de execução com diferentes valores de N deverão ser apresentados em um gráfico do tipo:

- (eixo y) tempo de execução ou número de passos (menor valor, maior valor e média)  
(eixo x) o tamanho do vetor N a ser ordenado

Lembre-se que os valores de N devem ser escolhidos de maneira que o gráfico permita tirar conclusões sobre o perfil de complexidade dos algoritmos: constante, logarítmico, linear, nlogaritmico ou quadrático.

- Só isso?

É! E o resto é igual antes.

- Ahhh, e aquele pontinho extra?

Se os gráficos forem apresentados utilizando alguma biblioteca de visualização científica (ex.: gnuplot), você vai ganhar uns pontinhos a mais.

Se você realizar um ajuste de curvas aos dados e incluir esta informação nos gráficos, você vai ganhar muitos pontinhos a mais.

Se você fizer uma análise estatística que demonstre numericamente qual a melhor classe de modelos explica a complexidade do algoritmo (ex: utilizando testes como likelihood-ratio), você ganhará uma enormidade de pontinhos a mais.

- E é para quando?

Entrega para o dia 29 de junho, via sigaa. Submeter o código e um relatório com os gráficos.

- E os grupos?

Vocês podem formar grupos de até 4 pessoas. Todos devem submeter individualmente via sigaa. Identificar os grupos no relatório.