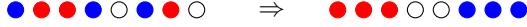


Estudo comparativo de métodos para o *Dutch National Flag*

Dutch National Flag: dado uma fileira de bolas pintadas de vermelho, branco ou azul, rearanjá-las de tal forma que as de mesma cor estejam juntas e os grupos de cores estejam na ordem da bandeira da Holanda.

Ex.: 

O problema foi proposto por Dijkstra (sim, aquele do caminho mínimo), cientista da computação holandês.

Uma versão do *Dutch National Flag* para um array A de números reais consiste em reordenar os números de tal forma que todos os negativos fiquem antes de todos os zeros e todos os positivos depois dos zeros.

Ex.: $0, \textcolor{blue}{1.618}, \textcolor{blue}{3.14}, \textcolor{blue}{2.718}, \textcolor{red}{-1.41}, \textcolor{blue}{0.577}, 0, 0, \textcolor{red}{-9.8}, 0 \Rightarrow \textcolor{red}{-1.41}, \textcolor{red}{-9.8}, 0, 0, 0, 0, \textcolor{blue}{3.14}, \textcolor{blue}{0.577}, \textcolor{blue}{1.618}, \textcolor{blue}{2.718}$

É possível resolver o problema ordenando-se todos os elementos de A em ordem crescente. Mas isso pode ser um desperdício de tempo, já que os números em cada grupo não precisam estar em ordem (veja o exemplo).

Neste trabalho você deve analisar e comparar dois métodos que resolvem o *Dutch National Flag* para array de números reais. Especificamente, estudar a influência de vários fatores no tempo gasto para obter o resultado: o método, o tamanho do vetor, a linguagem de programação, o estado inicial do array e a quantidade de zeros. São 5 fatores e alguns deles podem ter vários níveis, mas o estudo deve considerar apenas 2 níveis de cada:

- Método: de Dijkstra¹ e outro à sua escolha (ordenação total ou proponha um antes de ver o de Dijkstra!)
- Tamanho do array: em torno de milhões (é esperado que quanto maior mais demorado)
- Linguagem de programação: à sua escolha
- Estado inicial do array: array com dados aleatórios e quase ordenado² (é esperado que demore mais para o aleatório)
- Quantidade de valores: uniforme (\pm mesma quantidade de negativos, zeros e positivos) e com bem menos zeros que os outros.

1. Fatorial 2^4

Faça um estudo da influência dos fatores por um planejamento 2^4 , isto é, um fatorial completo com 4 fatores e 16 experimentos. Para isso, escolha 4 fatores que você acha que mais influenciam o resultado e use um nível fixo para o fator restante (a influência dele não será estudada nesta parte). Cada experimento é uma combinação de níveis dos fatores escolhidos e o resultado é o tempo gasto na execução. Por exemplo, um experimento pode ser marcar o tempo de execução do método de Dijkstra implementado em C++ com array aleatório de 1 milhão de itens com poucos zeros.

Monte a tabela de sinais do 2^4 , execute cada um dos 16 experimentos, faça a computação de efeitos e alocação da variação. Informe quais fatores ou interações mais influenciam o tempo e comente os resultados encontrados (o que resulta como esperado, se houve alguma descoberta interessante, etc).

Justifique a escolha dos fatores, já que nem todos poderão ser estudados.

¹Dijkstra propôs o problema e um método eficiente; procure por *Dutch National Problem* na Wikipedia para um pseudocódigo

²pode construir um iniciando com um array ordenado e modificando aleatoriamente o local de várias elementos ou movendo cada elemento para uma certa distância limitada à frente ou atrás

2. Fatorial 2^{5-2}

Faça um estudo da influência dos fatores por um planejamento 2^{5-2} , isto é, com 5 fatores e um total de 8 experimentos. Como no anterior, cada experimento é uma combinação de níveis dos fatores e o resultado é o tempo gasto na execução.

Monte uma tabela de sinais do 2^{5-2} , execute cada um dos 8 experimentos, faça a computação de efeitos e alocação da variação. Informe quais fatores ou interações mais influenciam o tempo e comente os resultados encontrados (o que resulta como esperado, se houve alguma descoberta interessante, etc).

Justifique a escolha de quais interações aparecem na tabela (já que nem todas poderão ser estudadas).

3. Fatorial 2^4r ou $2^{5-2}r$

Os estudos anteriores foram feitos com apenas uma execução de cada experimento. Entretanto, o resultado pode ser diferente para entradas ligeiramente diferentes. O tempo de execução pode ser diferente até para um mesmo array de entrada, dependendo de outros fatores.

Escolha um dos estudos anteriores e faça r replicações de cada experimento, ou seja, um fatorial completo 2^4r ou fracionado $2^{5-2}r$ com r replicações, sendo r um valor que achar apropriado ($r \geq 5$).

Faça a computação dos efeitos e a alocação da variação (inclusive dos erros experimentais). Calcule intervalos de confiança para os efeitos encontrados e faça uma estimativa do tempo médio de execução para pelo menos dois dos experimentos, considerando um número m (à sua escolha) de replicações. Em seguida, execute m novas replicações desses dois experimentos e informe se a média do tempo realmente ficou dentro do intervalo de confiança calculado.

Entregue um relatório em formato PDF contendo, pelo menos:

- introdução, com objetivos do trabalho e resumo do que foi feito e o que foi encontrado
- descrição (pseudocódigos ou códigos) dos métodos usados (*obs.: você pode propor e implementar seu código ou usar um pseudocódigo ou código disponível dos métodos – nesse caso, informar a fonte*)
- as tabelas de sinais dos planejamentos, os resultados dos experimentos e os valores calculados
- discutir os resultados, informando os fatores que causam maior variação, as interações, etc
- discutir as diferenças entre os projeto de experimentos realizados, suas vantagens/desvantagens etc
- informar/justificar as decisões tomadas, por ex., níveis escolhidos, colunas usadas na tabela, etc
- conclusão sobre o que aprendeu, tanto dos métodos e fatores quanto do projeto de experimentos
- se feito em grupo, indicar as atividades atribuídas a cada participante

Observações:

- a pontuação depende não apenas da quantidade ou relevância dos resultados analisados e das conclusões retiradas, mas também da qualidade e organização do relatório
- entregue, separado do PDF do relatório, os códigos usados em um arquivo compactado formato zip
- para evitar elementos repetidos, usar array de números de ponto flutuante em vez de inteiros