

INF 112 – Prática 3, Avaliação de desempenho – 2019/2

Exercício 1 O programa abaixo foi feito para avaliar a frequência com que cada número aparece em uma série de uso do método `rand() % limiteRand`.

```
int main(int argc, char **argv) {
    srand(time(NULL));

    int numTestes, limiteRand;

    if (argc != 3) {
        cerr << "Erro, use: ./exercicio numTestes limiteRand" << endl;
        exit(1);
    }

    numTestes = atoi(argv[1]);
    limiteRand = atoi(argv[2]);

    int *vetor = new int[numTestes];

    for(int i = 0; i < numTestes; i++)
        vetor[i] = rand() % limiteRand;

    for(int i = 0; i < limiteRand; i++) {
        int ct = 0;
        for(int j = 0; j < numTestes; j++)
            if (vetor[j] == i)
                ct++;
        cout << i << " " << ct << endl;
    }
    delete []vetor;
    return 0;
}
```

Ao executar o programa passando como parâmetro pelo terminal os valores 10 e 20, por exemplo, o computador sorteia 10 números entre 0 e 19 e, então, exibe um histograma falando quantas vezes cada número (entre 0 e 19) apareceu no sorteio (espera-se que os números sejam bem parecidos pois em um gerador de números pseudo-aleatórios é desejado que cada número tenha a mesma probabilidade de ser sorteado).

Compile o programa (como exercício) e faça o seguinte:

1. Execute o comando `time ./exercicio 1000 200 > saida.txt` (como a saída do programa é grande, vamos redirecioná-la para o arquivo saída) e veja quanto tempo o programa gasta para ser executado. O comando `time` do linux pode ser utilizado para medir o tempo de execução de um programa: o tempo “real” indica o tempo “medido no relógio” (também conhecido por “wall time”); o tempo “user” indica o tempo de uso da CPU e o tempo “sys” indica o tempo utilizado com chamadas ao sistema operacional.
2. Execute novamente o comando passando como parâmetro os valores 10000 e 10000
3. Execute novamente o comando passando como parâmetro os valores 10000 e 20000
4. Execute novamente o comando passando como parâmetro os valores 20000 e 10000
5. Execute novamente o comando passando como parâmetro os valores 30000 e 30000
6. Execute novamente o comando passando como parâmetro os valores 100000 e 30000
7. Faça outros testes.
8. Pergunta: com base nos resultados obtidos nos testes, como você acha que os parâmetros do programa influenciam no tempo de processamento?
9. Analisando o código, qual a parte do programa você acha que “pesa” mais no tempo de processamento?
10. Modifique o programa de forma que ele fique (BEM) mais eficiente (dica: o código pode ser muito melhorado modificando-se a ideia do algoritmo).
11. Após melhorar o programa, troque o `endl` por um “`\n`” e, antes do `return 0;`, coloque um `cout << flush`. Veja a influência que isso tem no tempo de processamento. Explique tal influência.
12. Finalmente, ao compilar a última versão do programa, adicione a flag “-O3” às diretivas do g++. Teste o programa novamente e explique a diferença de tempo.