



SCC0220 - Laboratório de Introdução à Ciência da Computação II

Prof. Jean R. Ponciano

Estagiário PAE: João Victor C. N. de Sousa

Monitores: Matheus Vieira Fernandes e Fernando Valentim Torres

Departamento de Ciências de Computação (SCC)
Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo

Entregável 05 – Tiro no escuro

Dessa vez, o professor ficou sem criatividade para criar uma história. Este trabalho pede “apenas” que uma sequência com os **N** primeiros valores inteiros ($0 < N < 10^9$) seja ordenada. O único problema é que a escolha do algoritmo de ordenação que vai passar em todos os casos de teste do run.codes é um tiro no escuro que você precisa dar, já que a sequência a ser ordenada pode ser tal que programas com certos algoritmos de ordenação podem não conseguir rodar dentro do tempo estipulado ou dar falha de segmentação. Trabalhe com as entradas e saídas no formato abaixo e capriche nos seus tiros! Pelo menos três deles precisam acertar o alvo.

Exemplos de entrada

```
100000  
reverse  
2
```

```
100000  
random  
1
```

```
100000  
sorted  
3
```

A primeira linha diz o valor de N. Isso indica que a sequência irá de 1 a N.

A segunda linha informa como a sequência deve ser considerada na entrada.

- “reverse” (note que o input não possui aspas) diz que a sequência de entrada é apresentada ao algoritmo de ordenação com os N elementos em ordem inversa (i.e., [N, N-1, N-2, ..., 1]).

- “random” diz que a sequência de entrada é apresentada com N elementos aleatórios entre 1 e N (inclusive com repetições).
- “sorted” diz que a sequência de entrada é apresentada com os N elementos já ordenados ([1, 2, ... N-1, N]).

A terceira linha informa o id do algoritmo de ordenação a ser chamado. **Seus três melhores algoritmos devem ter ids 1, 2, 3, e eles devem passar nos casos de teste 1, 2 e 3, respectivamente, existentes no run.codes.** Outros algoritmos eventualmente implementados devem ter id > 3, e não há casos de teste para eles.

Exemplos de saída

```
96C4337A
```

```
487358B9
```

```
96C4337A
```

A saída do seu programa deve ser o hash da sequência ordenada de acordo com o [CRC32](#). Wtf? Essa é a melhor maneira encontrada para que o run.codes possa trabalhar com a ordenação de grandes sequências sem que haja grandes entradas e saídas.

Muita calma nessa hora. Você não precisará implementar essa técnica de hashing. Basta:

1. Pegar o util.h lá no run.codes, junto ao enunciado do exercício.
2. Incluir no cabeçalho do seu código principal

```
#include "util.h"
```

3. E gerar a saída do seu código usando:

```
init_crc32();
uint32_t saida = crc32(0, sequencia_ordenada, N * sizeof(int));
printf("%08X", saida);
```

Outro ponto importante: Quando a segunda linha for random, é preciso garantir que a saída do seu programa (que depende de números aleatórios) seja igual à esperada no run.codes. Por isso, é importante que você gere seus números aleatórios usando:

```
int seed = 12345; //Não altere esse valor.
int numero_aleatorio = get_random(&seed, N);
```

A função `get_random` está implementada em `util.h` e utiliza o gerador de números aleatórios [Xorshift](#) (x32), que independe de plataforma e vai gerar os mesmos números no `run.codes` e na sua máquina (desde que o valor de `seed` seja igual).

Submissões:

O trabalho completo deve conter a implementação de, pelo menos, os três algoritmos de ordenação (diferentes ou variações) que passem em todos os casos de teste e um relatório.

1. `Run.codes`: O código submetido deve ter as implementações de todos os algoritmos necessários para passar em todos os casos de teste, com o usuário podendo escolher o algoritmo a ser executado via teclado (relembre a terceira linha da entrada).
2. E-disciplinas: Relatório em PDF **com até cinco páginas** contendo uma breve descrição de como cada algoritmo funciona, além de avaliação comparativa de tempo de execução entre as versões e discussão sobre suas complexidades computacionais. Dentre outras, responda: Por que esses três algoritmos passaram nos testes e por que os outros não? Dentre os três, qual foi mais rápido/mais lento e por que isso aconteceu?

Prazo: até dia 23/10

Atenção:

Caso haja suspeita de uso de IA, o professor poderá requisitar apresentação e arguição sobre o que foi submetido. Neste caso, a nota (individual) será dada a partir do desempenho obtido na apresentação/arguição.