

Trabalho 7 - De volta às raízes

Joãozinho está tendo problemas ao organizar seu baralho, ajude-o a organizar mostrando passo a passo da organização.

Implementação

Implemente os métodos de ordenação Radix Sort e Stoogesort, você receberá $0 < k < 10^6$ cartas.

Joãozinho tem um baralho único e especial, o baralho contém cartas com naipes (\spadesuit \heartsuit \clubsuit \diamondsuit , nesta ordem) e $0 < n < 150$ valores. A sequência de valores, do menor ao maior, é: 4, 5, 6, 7, Q, J, K, A, 2, 3. A ordenação deve ser lexicográfica, considerando primeiro o naipe.

Radix Sort:

- Use a variante LSD(dígito menos significativo)
- Mostre cada passo da ordenação no formato: `<carta-1>;<carta-2>;...;<carta-n>;`
- Para cada passo, imprima: “Após ordenar o i° dígito dos valores:” (substituindo i pelo dígito atual)
- Ao final, imprima: “Após ordenar por naipe:”

Stoogesort

Apresente apenas o resultado final da ordenação, sem o passo a passo.

Exemplo

Entrada:

4 3

\spadesuit 456

\heartsuit 4JK

\clubsuit AAA

\diamondsuit 532

Saída:

\heartsuit 4JK; \spadesuit 456; \clubsuit AAA; \diamondsuit 532;

Após ordenar o 3° dígito dos valores:

\spadesuit 456; \heartsuit 4JK; \clubsuit AAA; \diamondsuit 532;

Após ordenar o 2° dígito dos valores:

\spadesuit 456; \heartsuit 4JK; \clubsuit AAA; \diamondsuit 532;

Após ordenar o 1° dígito dos valores:

\spadesuit 456; \heartsuit 4JK; \diamondsuit 532; \clubsuit AAA;

Após ordenar por naipe:

\spadesuit 456; \clubsuit AAA; \heartsuit 4JK; \diamondsuit 532;

Relatório

- Compare os métodos de ordenação e explique brevemente como o Radix Sort consegue atingir uma complexidade linear.
- Além disso, demonstre qual caso do run.codes demora mais de um minuto para ser concluído na implementação do Stoogesort.
- Conclua explicando por que o Radix Sort é adequado para ordenar cartas.

Dicas:

É recomendado o use de strings para representar os naipes, mas se quiser se aventurar com `wchar_t`, fique à vontade!