Relatório EP3

**Como funciona seu algoritmo? Qual a ideia geral?**

A ideia geral é que uma sequência de panquecas (números) possui sub-regiões já organizadas, tendo consciência de quais são esses “blocos”, não devemos flipar a torre de panquecas no meio dessa região, para não a desorganizar. Devemos nos preocupar apenas com os extremos a serem flipados, portanto.

Quando acontece um flip em uma posição “i”, todas as panquecas entre a posição “i” e “tam – 1” são giradas, mas o que importa é o que acontece com as pontas desse grupo de panquecas (a de cima vai para a posição da de baixo e vice-versa).

Se soubermos, então, onde está a panqueca seguinte ou anterior à do topo da torre, podemos encaixar as duas pontas sem desorganizar os blocos dentro do intervalo entre elas.

Para saber qual deve ser a organização correta, criamos um vetor auxiliar em que as panquecas são postas e organizadas pelo algoritmo “bubblesort”. Assim que estiverem organizadas, pode-se fazer uma correspondência entre a posição das panquecas no vetor auxiliar e a delas no vetor principal, graças ao uso de um struct panqueca, que guarda 3 inteiros (o valor da panqueca, a posição dela no vetor principal e a cardinalidade do valor da panqueca, ou seja, a posição em que ela deveria estar num vetor em que essas panquecas estão todas organizadas.

**Qual sua estimativa de complexidade de tempo de seu algoritmo para ordenar n panquecas?**

Considerando um caso em que não qualquer bloco formado na torre de panquecas, e considerando o pior caso (quando é preciso dar 3 flips para juntar duas pontas, ou seja, quando o topo da torre precisa se juntar com uma panqueca que está contida num bloco organizado imediatamente à direita dessa panqueca), o tempo necessário para ordenar n panquecas é limitado por 3 \* n, ou seja, tem complexidade O(n).

**Qual o número mínimo e máximo de flips seu algoritmo faz para ordenar uma sequência?**

O número mínimo de flips que o algoritmo dá é 0, pois a função “onde\_flipa” inicia verificando se a torre de panquecas está organizada. Um vetor já organizado não precisa ser flipado.

O número máximo de flips que o algoritmo executa é 3 \* n, como informado na resposta à pergunta anterior.

**Mostre sequências em que seu algoritmo realiza o número máximo e mínimo de flips.**

Legenda: (n) / sequência de “n” números

Onde o número de flips é mínimo:

(3) / 3 2 1

(10) / 25 19 17 13 10 8 6 4 3 1

(8) / 289 256 225 196 169 144 121 100

(10) / 89 45 23 18 11 5 4 3 2 1

Onde o número de flips é máximo:

(7) / 300 10 250 55 280 44 120

(8) / 7 8 10 12 13 1 2 5

(5) / 1 4 3 8 7

(18) / 8 8 8 8 10 10 12 8 10 12 12 10 8 12 12 12 12 10

**Mostre sequências em que seu algoritmo não realiza o número mínimo possível de flips para ordenar uma sequência.**

(10) / 8 5 9 6 3 2 4 4 5 5

(15) / 8 8 9 7 7 5 4 8 8 0 2 10 10 8 8

**Você consegue alguma estimativa para a qualidade do seu algoritmo, ou seja, por exemplo, o número de flips dado por seu algoritmo é limitado por uma constante vezes o número ótimo de flips? Ou, o número de flips do seu algoritmo é limitado por n (número de panquecas) vezes o ótimo?**

Como dito anteriormente, o algoritmo é limitado pelo triplo da quantidade de panquecas da torre.

**Imagine uma versão do problema em que as panquecas tenham um lado mais queimado que deva ficar voltado para baixo. Seu algoritmo funciona neste caso?**

O algoritmo não necessariamente funcionaria nesse caso, porque ele dependeria de como cada panqueca está virada no começo da organização e de quantos flips são dados nas subpartes da torre durante esse processo.