



Universidade de Brasília
Departamento de Ciência da Computação
Projeto e Análise de Algoritmos

Atividade 08

José Antônio Alcântara da Silva de Andrade Mat: 232013031

Professor:
Flávio Leonardo Calvacanti de Moura
Turma 02

Brasília, DF
3 de junho de 2025

Exercício

“Mostre como podemos ordenar n inteiros contidos no intervalo de 0 a $n^2 - 1$ em tempo linear, ou seja, $O(n)$.”

Resolução

Para a resolução desse exercício, consideraremos o uso da base decimal para a representação dos números. Os conceitos, porém, estendem-se para outras bases, como veremos.

Considere dez filas (estruturas de dado FIFO), cada qual com a nomenclatura de um dos dez dígitos da base decimal. Chamaremos cada uma de “balde”, ou seja, teremos o **balde0**, **balde1**, e assim por diante até o **balde9**. Teremos também uma outra fila, que denotaremos por **mainf**.

Iniciando-se o algoritmo, coloca-se a ordenação inicial dos números na fila **mainf**. Em seguida, seguindo a ordem da fila, checka-se o dígito da unidade do primeiro termo. Coloca-o no balde correspondente. Realiza-se tal processo repetidamente, até que **mainf** esteja vazia, e cada número esteja no seu balde correspondente, de acordo com seu dígito das unidades e sua posição de entrada no balde.

Em seguida, retira-se os números do **balde0** e os retorna à fila **mainf**, em ordem. Faz-se o mesmo para cada balde, também em ordem: **balde1**, **balde2**, ..., **balde9**. Dessa forma, teremos que a fila **mainf** estará ordenada em função do valor do dígito das unidades de cada número.

Esse processo é repetido para cada dígito dos números, preenchendo lacunas com zeros à esquerda quando necessário (o número 19 viraria 0019, por exemplo). Após o término do processo, os números estarão ordenados. Note que não ocorre comparações entre itens nesse algoritmo.

O algoritmo apenas funciona devido ao fato que a ordem entre itens de mesma classe (do mesmo “balde”) não é alterada. Por exemplo, suponha que na ordenação das unidades, um certo item A foi classificado como do **balde1**, e um outro item B do **balde4**. Em seguida, na ordenação das dezenas, encontrou-se que A e B ambos pertencem ao **balde5**. Acontecerá que, como A pertencia à classe menor, ele será analisado antes de B , garantindo que a ordem permaneça do menor ao maior.

Para a análise desse algoritmo, note que a quantidade de “baldes” influenciará a velocidade. Analisando primeiramente com os dez baldes, precisamos considerar a quantidade de vezes que o algoritmo realizará seu ciclo principal. Isso depende do maior número, ou seja, a quantidade de iterações $d = \lceil \log_{10} N \rceil$ onde N é o maior número da lista. Em seguida, nota-se que, para cada número de **mainf**, realiza-se uma comparação estática, ou seja, $O(1)$. Como são n elementos, teremos então que cada iteração demanda n operações. Assim, teremos que a complexidade desse algoritmo é $O(d \cdot n) = O(n)$.

Pode-se generalizar o valor de d ao notar que a quantidade de baldes aparece como o valor da base do logaritmo. Ou seja, escrevemos $d = \lceil \log_b N \rceil$ onde b é a quantidade de baldes (as classes diferentes de valores) e N o maior número presente na lista.