Universidade Estadual do Oeste do Paraná – UNIOESTE
Campus de Cascavel
Colegiado de Ciência da Computação

Projeto e Análise de Algoritmos
Lista 3 – 28/10/2022

1) Com base nas características dos métodos de ordenação vistos em sala, assinale, dentre as afirmações apresentadas a seguir, a alternativa correta. Considere, para tanto, que se deseja ordenar o conjunto de forma crescente.
a) O pior caso para o algoritmo MergeSort dá-se quando o vetor de entrada está ordenado de forma decrescente.
b) O pior caso de complexidade assintótica para o Quicksort ocorre quando o elemento selecionado como pivô está exatamente no centro (mediana) do conjunto.
c) O método SelectionSort faz menos comparações quando o conjunto de entradas já está ordenado de forma crescente.
d) O algoritmo Quicksort fará no mínimo n operações de swaps em um conjunto de tamanho n.
e) O BubbleSort realiza o menor número de operações de swap quando o conjunto de entrada está ordenado de forma decrescente.

2) Considere o algoritmo a seguir.

```
MERGESORT(V, i, j)
(1) Se (i<j) então
(2)       m = (i+j)/2;
(3)       MERGESORT(v, i, m);
(4)       MERGESORT(v, m+1, j);
(5)       MESCLAR(v, i, m, j);
(6) Fim;
```

Sobre o comportamento assintótico do algoritmo de ordenação *Merge Sort*, assinale a alternativa que apresenta, corretamente, sua complexidade.
a) O(log n)
b) O(n log n)
c) $O(n^2)$
d) $O(n^3)$
e) O(2n)

3) Um pesquisador visando encontrar um método de ordenação eficiente criou uma nova abordagem para o algoritmo mergesort, chamando-o de supermerge. A ideia do algoritmo é similar ao original, porém ao invés de dividir o conjunto de entrada em duas partes, a divisão é feita obtendo-se quatro partes. Ajude o pesquisador mostrando se a abordagem por ele proposta, em termos de análise assintótica, será mais eficiente que o método original. Para fazer tal demonstração, utilize o método da substituição.
Supermerge terá complexidade $O(n \log_4 n)$
$n \log_4 n = 2\ n \log_2 n$ logo podemos dizer que eles são equivalentes, já que o 2 pode ser cortado por ser constante...

4) Marque V para as afirmações verdadeiras e F para as falsas.
(F) Dependendo da distribuição dos dados em um conjunto o Bubblesort pode ter custo linear.
(V) Ao se adotar o Bucketsort em conjunto com o SelectionSort é possível diminuir o custo de execução de um algoritmo de ordenação mas não seu custo assintótico.
(V) O método de ordenação por contagem é bastante interessante quando os dados estão bem espalhados ao longo do espaço entre o menor e maior valores presentes no conjunto.
(V) O algoritmo de ordenação por seleção pode chegar a uma complexidade linear dependendo da distribuição dos valores do conjunto.

5) Por que o MergeSort é considerado um método de ordenação de custo constante?
Ele é considerado constante pois independe da distribuição dos elementos no conjunto a ser ordenado. Se o vetor estiver todo bagunçado, já ordenado ou decrescente, o algoritmo não analisa isso, ele simplesmente divide o conjunto em segmentos e depois vai agrupando-os de forma ordenada...

6) Demonstre, matematicamente, o ganho ao se adotar o método de ordenação por baldes empregando-se o algoritmo SelecionSort para ordenar cada balde em comparação à estratégia de ordenação por seleção sozinha. Para tanto, considere a adoção de 10 baldes.

Selection clássico (polinômio simplificado): $O(n^2)$
Considerando um n=100 teríamos um custo de 100^2=10000 operações

Selection usando bucket (usando 5 baldes): $O(n) + O((n/5)^2)+O(n)$
Considerando o mesmo n=100 teríamos um custo de $100+20^2+100$=600 operações

7) Dado o vetor a seguir, qual seria a sequência de seleção dos pivôs mais interessante dentro do algoritmo de Quicksort? E qual seriam a pior sequência de escolha dos três primeiros pivôs?

7	6	13	2	9	11	4	8	3	1	19	20	14	17	5	12

O melhor pivô para o quicksort é aquele que representa a mediana do conjunto, de forma que o segmento à esquerda do pivô tenha a mesma dimensão do segmento à direita dele. Para tanto, as melhores escolhas seriam o 8 ou 9. Em seguida, no segmento da esquerda os melhores seriam o 4 ou 5 enquanto do lado direito seriam os valores 13 ou 14. Esse processo continua iterativamente...

As piores escolhas de um pivô seriam os elementos das pontas do vetor. Por exemplo, selecionar os valores 1 ou 20 na primeira iteração, os valores 2 ou 19 na segunda iteração, os valores 3 ou 17 na terceira iteração...

8) O que aconteceria se no momento de executar o BucketSort os elementos do conjunto fossem quase todos colocados no mesmo balde? Explique através de um exemplo.

Nesse caso o algoritmo teria um aumento no custo temporal em relação à solução inicial, ou seja, seria pior usar o bucket do que não o usar. No entanto, o custo assintótico permaneceria o mesmo.

Considerando o vetor da questão 7 teríamos o custo de percorrer o vetor inteiro para descobrir em que balde cada elemento seria posto. O que geraria um custo O(n). Em seguida teríamos o custo cheio de execução do método de ordenação (insertionsort por exemplo). Por fim, teríamos o custo de concatenar os elementos de todos os baldes, o que geraria um custo adicional de O(n). Assim o custo total seria O(n) + InsertionSort(n) + O(n), que é pior que o custo InsertionSort(n). Nos dois cenários o algoritmo teria $O(n^2)$ como pior caso.

9) Em que caso o InsertionSort pode incorrer em uma complexidade linear? Explique a partir do algoritmo, a razão deste fato ocorrer.

Quando o vetor já estiver ordenado o custo do algoritmo será linear. Esse fato ocorre pois a condição Vet[j]>aux do controle do while do algoritmo (linha 6 do exemplo visto em sala) nunca será verdadeira. Isso faz com que ele nunca entre no laço interno do método, mantendo-se em complexidade linear.