Lista 6

Caio Ian Rozendo Silva de Melo

Métodos Computacionais

UFPE
Brasil
March 11, 2021

1

```c
// Função para trocar os valores dados
void swapi(int *a, int *b) {
        int c = (*a);
        (*a) = (*b);
        (*b) = c;
}
```

a)

```c
// Ordenação de um vetor de inteiros por seleção, onde r = N
void ord_selecao(int *v, int r) {
        int i, j, s;

        for(i = 0; i < r; ++i) {
                for(j = 0; j < r; ++j) {
                        if(v[i] < v[j] && i < j)
                        ↪  swapi(&v[i], &v[j]);
                }
        }
}
```

Independente do vetor estar parcialmente ordenado, cada elemento faz $N^2 = 64$ comparações(podendo ser marginalmente melhorado se "pularmos" a comparação com o elemento do mesmo índice). Já a quantidade de trocas $T = 2$

b)

```c
// Ordenação de um vetor de inteiros por inserção do índice
↪  (max, 0), onde max = N e b é um booleano para garantir
↪  que apenas a primeira recursão varrerá o vetor inteiro.
void ord_ins(int *v, int r, int max, int b) {
        int s;

        if( (r - 1) >= 0 && (v[r-1] > v[r]) ) {
                swapi(&v[r-1], &v[r]);
                if(r+1 < max) ord_ins(v, r+1, max, 1);
        }
        if(b == 0 && r > 0) ord_ins(v, r-1, max, b);
}
```

Foram feitas $C = 9$ comparações e $T = 2$ trocas para o vetor dado.

2

a) Inicialmente, um vetor de $N = 100$ é inicializado com valores aleatórios com valores compreendidos entre $[0; 10 \cdot N)$, após isso envia-se o vetor para ser ordenado por seleção, onde faz-se uma cópia na memória dos valores do vetor anterior; dentro do laço, à cada iteração, procura-se a posição no vetor no menor índice em relação ao pivô na posição "i" e caso ínfero ou igual, faz-se a troca entre eles; no final soma-se a quantidade de comparações com o valor prévio e repete-se até o valor $N - 1$. Após a rotina, imprime-se o vetor ordenado resultante. Nas inserções, faz-se ordenamento por inserções(com outra cópia do vetor inicial) que também percorre o vetor de forma linear, porém, com menos comparações pois não é necessário verificar valores anteriores a não ser que hajam trocas de posições.

d) A quantidade de comparações feitas entre a ordenação por inserção é menor que a de seleção, porém, com algumas melhoras podemos diminuir o número de comparações na seleção, sendo no algorítmo dado igual a $T = \frac{N(N+1)}{2}$ e procurando-se pelo menor índice nos gera N trocas.