



UNIVERSIDADE DE BRASÍLIA
Organização e arquitetura de computadores

Relatório Laboratório 1

Grupo 3

Davi Mansur Costa - 211042701
Hagatta Giovanna Nogueira de Sousa - 190088508
Leonardo Ramos Barbosa - 211010360
Maria Eduarda Alves de Sousa - 190113219
Pedro Brum Tristão de Castro - 202067470

Respostas:

1.2) Considere a execução deste algoritmo em um processador RISC-V com frequência de clock de 50MHz que necessita 1 ciclo de clock para a execução de cada instrução (CPI=1). Para os vetores de entrada de n elementos já ordenados $Vo[n] = \{1, 2, 3, 4, \dots, n\}$ e ordenados inversamente $Vi[n] = \{n, n-1, n-2, \dots, 2, 1\}$:

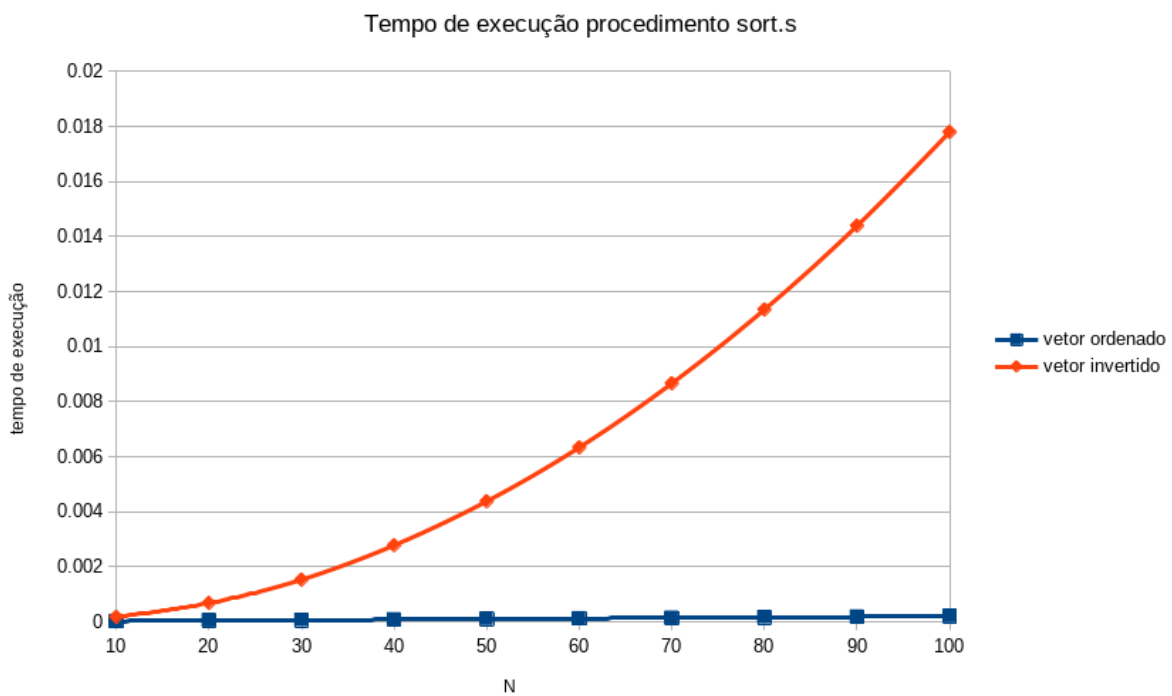
$$t_{\text{exec}} = N^{\circ} \text{ instruções} \times \text{CPI} \times T$$

a) Para o procedimento sort, escreva as equações dos tempos de execução em função de n , $t_o(n)$ e $t_i(n)$,

$$t_o(n) = 2.6 \cdot 10^{-6} + (2 \cdot 10^{-6})n$$

$$t_i(n) = (1.678 \cdot 10^{-6})n^{2.01866}$$

b) Para $n=\{10,20,30,40,50,60,70,80,90,100\}$, plote (em escala!) as duas curvas em um mesmo gráfico $n \times t$.



Parte B

2.2) Dado o programa sortc.c, compile-o com as diretivas -S -O0 e obtenha o arquivo sortc.s. Indique as modificações necessárias no código Assembly gerado para que possa ser executado corretamente no Rars.

As modificações feitas para compilar o código gerado utilizando a diretiva -S no RARS foram:

- Adicionar o .data antes da declaração do v.
- Adicionar o .text assim que se inicia o código.
- Trocar a função SHOW dada pelo Compiler Explore pela função SHOW definida no sort.s.
- Trocar o call printf para ecall SHOW.
- Trazer o *main* para o início do programa.
- Na *main* trocar a instrução jr ra para li a7,10 e logo depois um ecall.

2.3) Compile o programa *sortc_mod.c* e, com a ajuda do Rars, monte uma tabela comparativa com o número total de instruções executadas pelo programa todo, e o tamanho em bytes dos códigos em linguagem de máquina gerados para cada diretiva de otimização da compilação {-O0, -O1, -O2, -O3, -Os}. Compare ainda com os resultados obtidos no item 1.1) com o programa *sort.s* que foi implementado diretamente em Assembly. Analise os resultados obtidos usando o mesmo vetor de entrada.

Diretiva	Nº de instruções	Tamanho em bytes
-O0	9787	520
-O1	3891	364
-O2	2180	264
-O3	2175	380
-Os	4098	340

	Nº de instruções	Tamanho em bytes
sort.s	3740	272

3.1) Crie um procedimento *SORTEIO* que dado um número inteiro N (em $a0$), $2 \leq N \leq 20$, e um ponteiro C (em $a1$) crie aleatoriamente (ecall 41) na memória de dados, um conjunto $C = \{loja, c1, c2, \dots, cN\}$ de N casas de clientes cn em que cada casa é caracterizada pelas suas coordenadas (x, y) onde $x \in (0,310)$ e $y \in (0,230)$. A loja é a casa ($c0$) definida em (155,115). Exemplo de definição manual: $C: .word\ 155,115, 10,10, 10,20, 5,30$ # conjunto C de $N=3$ pares x,y

Arquivo fonte (.s) com o nome 3.1.s anexado à pasta.

3.2) Crie um procedimento DESENHA que, dado $N(a0)$ e $C(a1)$, desenhe na tela as posições das casas sorteadas usando a posição do pixel superior esquerdo de um quadrado verde de 8 pixels de lado contendo seu número como o caractere ASCII $(n + 65)$, conforme o exemplo abaixo. Desenhe a loja $c0$ com fundo vermelho. Filme a execução com $N=10$;

Arquivo fonte (.s) com o nome 3.2.s anexado à pasta.

3.3) Crie um procedimento ROTAS que, dado $N(a0)$ e $C(a1)$, desenhe as linhas que ligam uma a todas as outras casas e monte uma matriz $D=d_{ij}$ de dimensão $N \times N$ com suas distâncias Euclidianas (obs.: o motoboy anda onde quiser!). Filme a execução com $N=10$.

$$d_{i,j} = \sqrt{(c_i(x) - c_j(x))^2 + (c_i(y) - c_j(y))^2}$$

Arquivo fonte (.s) com o nome 3.3.s anexado à pasta.

3.4) Crie um procedimento ORDENA, que, dado $N(a0)$, $C(a1)$ e $D(a2)$, ordene o conjunto C e a matriz D , e desenhe em vermelho a linha que inicia em uma casa $c0$ (loja), une todas as casas, sem passar duas vezes em uma mesma casa, e volte à loja $c0$, de modo a ter menor distância total f percorrida possível, isto é: $f = \min \{ \sum_{i=0}^{N-1} d_{i,i+1} + d_{N,1} \}$. Filme as execuções para $N = \{2,3,4,5,6,7,8,9,10\}$

Arquivo fonte (.s) com o nome 3.4.s anexado à pasta.

Link para o vídeo dos testes: [‘UnB – OAC - Turma 03 - 2023/1’ – Grupo 3 - Laboratório 1- Problema do entregador de pizzas.](#)