

”

E-fólio B | Folha de resolução para E-fólio

UNIDADE CURRICULAR: Arquitetura de Computadores

CÓDIGO: 21010

DOCENTE: Gracinda Carvalho e José Coelho

A preencher pelo estudante

NOME: Pedro Pereira Santos

N.º DE ESTUDANTE: 2000809

CURSO: Licenciatura em Engenharia Informática

TRABALHO / RESOLUÇÃO:

Para todas as sub-rotinas, após as analisar, fiz um fluxograma para cada uma, dividindo assim cada sub-rotina em passos menos complexos e mais pequenos e fáceis de aplicar.

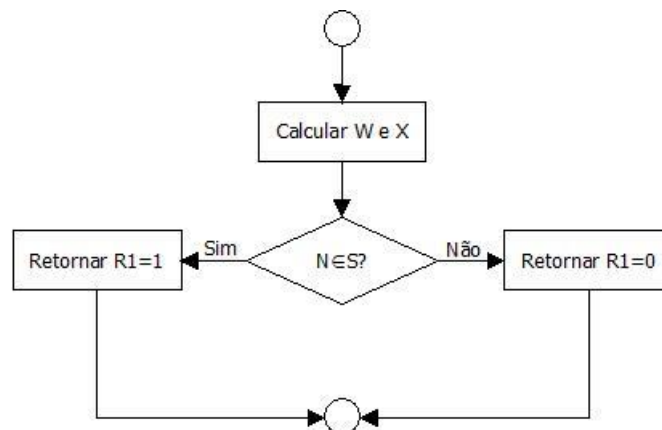
Com o objetivo de cada sub-rotina, adapta-se os passos intermédios ao melhor desempenho e funcionamento do programa, identificando, com a ajuda dos comentários, cada um, no próprio código.

Com a ajuda de comentários a acompanhar cada instrução no código, o seu desenvolvimento é facilitado com o aumento da legibilidade e maior compreensão do seu funcionamento.

Posto isto, temos a análise a cada alínea, começando com o fluxograma de cada uma.

Alínea A

Fluxograma TesteNM



No fluxograma W e X representam $(n \% m)^2 + 2 \cdot (n \% m)$ e $2 \cdot (n \% m) + 1$, respetivamente.

Depois de ter realizado o fluxograma podemos observar que o pedido é simples e direto, tendo só que ter o cuidado de copiar as variáveis necessárias uma vez que as operações de multiplicação e divisão são destrutivas.

Clock: 6933

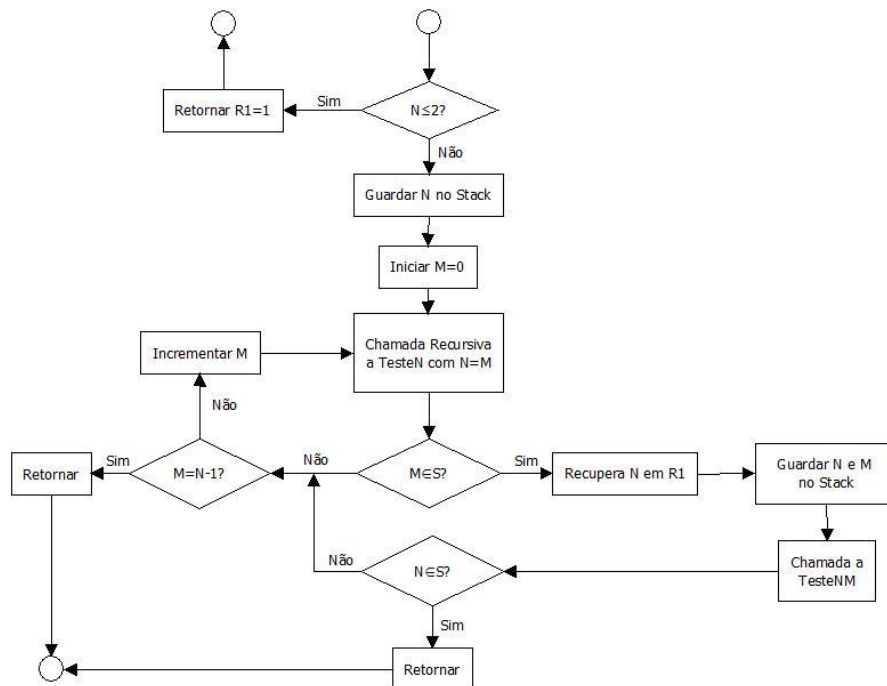
Instructions: 399

Alínea B

Após ter feito o primeiro rascunho do fluxograma para esta Alínea e indo escrevendo o código a acompanhar os vários passos do fluxograma, fui melhorando o fluxograma, juntando passos idênticos, diminuindo a complexidade e aumentando o desempenho da sub-rotina, ficando este fluxograma como o Final.

Para $N \leq 2$ retorna diretamente $R1=1$. Caso contrário decidi que o mais rentável a fazer era guardar o valor de N na stack para aceder mais tarde, quando tivermos um valor de M em que $M \in S$. Faz-se a chamada recursiva à sub-rotina para encontrar o valor de M em que $M \in S$, tendo $N=M$ para a nova chamada.

Fluxograma TesteN



Como para fazer a chamada a TesteNM é necessário enviar as variáveis N e M em R1 e R2, respetivamente, decidi que o melhor era guardá-las em stack para recuperá-las no retorno da sub-rotina.

Caso $N \in S$, $R1=1$, para informar que o valor pertence a S e retorna em R2 (Variável M) o valor de N para voltar a testar na chamada anterior, caso ela exista.

Caso não exista M tal que $N \in S$, retorna $R1=0$.

Clock: 4 062 845

Instructions: 301 715

Alínea C

Para fazer esta alínea fiz como estava a recomendar no eFólio e passei a alínea B à frente.

Após realizar o fluxograma, para implementar o pedido nesta alínea, decidi fazer um ciclo para que enquanto $N \leq 2$, guardar, na posição $N+1$ do vetor, $R1=1$, uma vez que se $N \leq 2$, $N \in S$.

Caso contrário, o melhor era criar um apontador auxiliar para percorrer todo o vetor verificando, segundo o valor do bit na posição $M+1$, se $M \in S$.

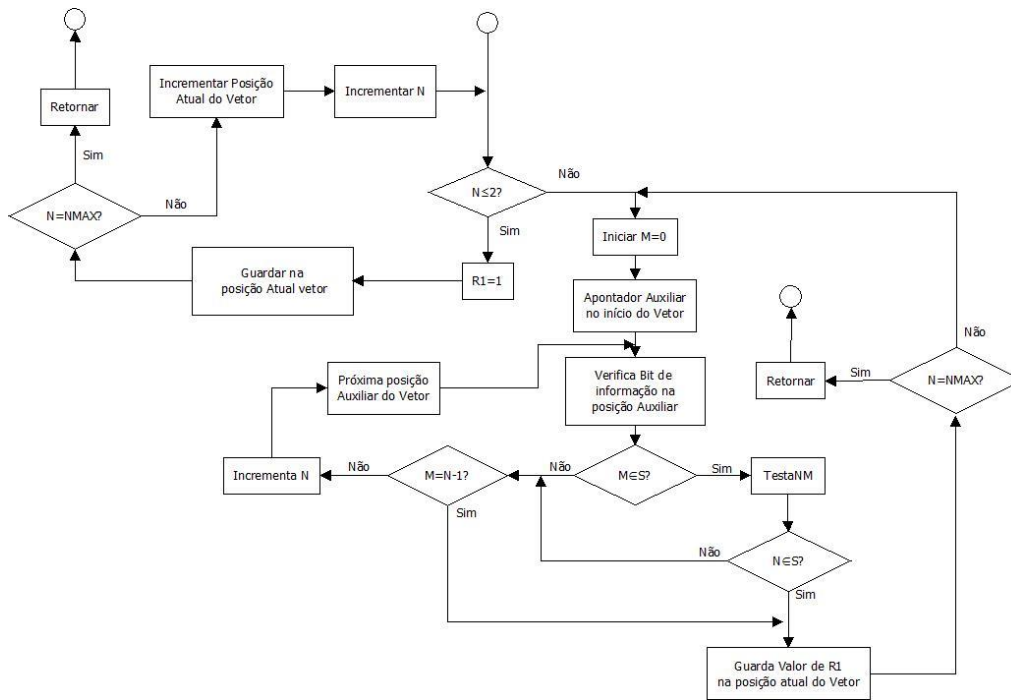
Após testar se $N \in S$ ou $M=N-1$ e $N \notin S$, guarda o valor corresponde de R1 na posição de memória $N+1$, acompanhado o incremento da posição de memória com o incremento da variável N.

Antes da chamada da sub-rotina TesteNM, para não perder os valores nos registos necessário à execução desta sub-rotina, decidi que o melhor era guardá-los na stack, para serem recuperados quando esta retornar.

Clock: 398 945

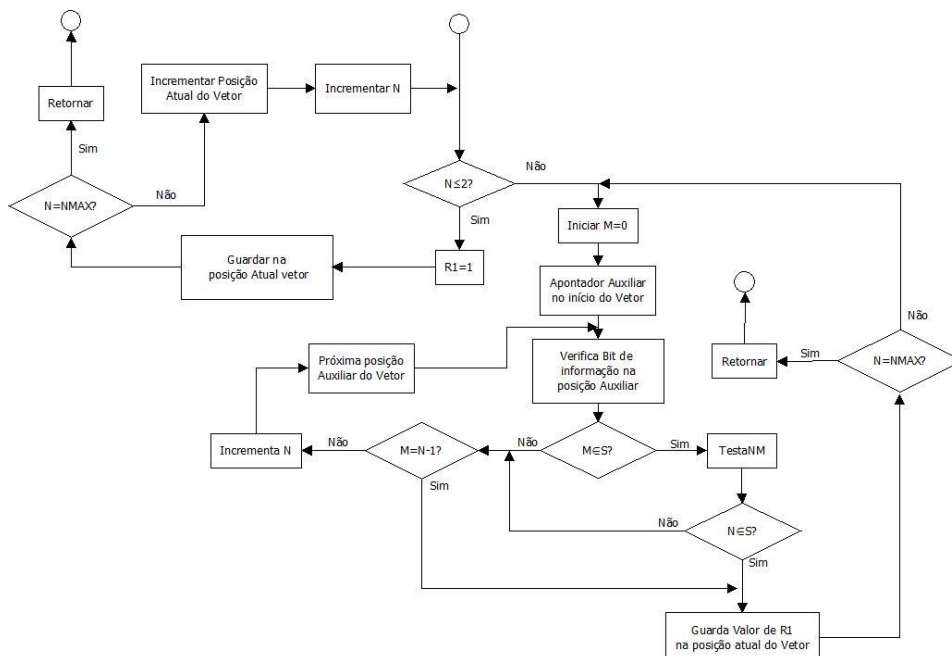
Instructions: 29 717

Fluxograma TesteVetorN

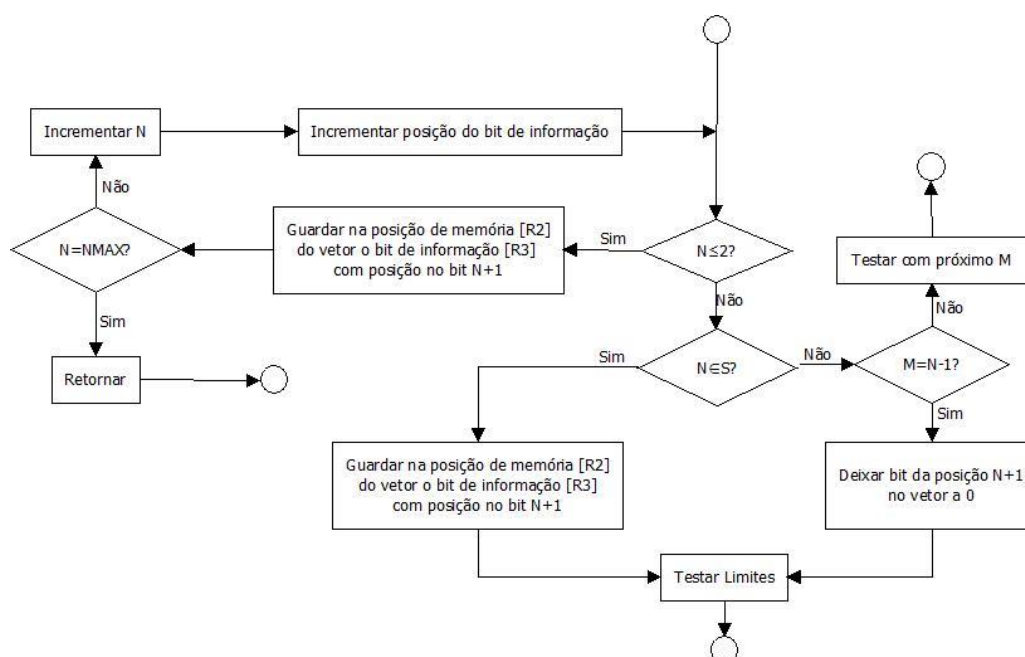


Alínea D

Fluxograma TesteVetorB



Fluxograma GuardarValorB



Para fazer esta sub-rotina, usei como base a anterior, sendo que o fluxograma é igual ao do **TesteVetorN**, modificando a parte de guardar os valores e de verificar se $M \leq S$.

A parte mais complicada que apanhei foi a de guardar os valores, necessitando de fazer um fluxograma, fluxograma **GuardarValorB**, para me ajudar no desenvolvimento dessa parte do código.

Depois, tanto para a leitura como para guardar o valor no vetor, após cada uma destas operações, sendo que não se passou os limites, tanto de N nem M , verificar se o próximo bit pertence à posição atual de memória, ou se é preciso incrementá-la.

Quando o valor guardado fica após a primeira posição de memória, é necessário iniciar o apontador auxiliar na primeira posição de memória, para compara os valores de M , mas como é necessário utilizar todos os registos no código desta sub-rotina, optei por guardar em stack a primeira posição do vetor, para quando for necessário reiniciar a variável auxiliar, é só recuperar o valor e copiá-lo para o registo encarregado dessa variável, uma vez que além de iniciar os apontadores, esse é o único momento para o qual é necessário.

Clock: 56 821 902

Instructions: 4 942 961

TOTAL (Todas as sub-rotinas juntas no programa teste)

Clock: 61 266 832

Instructions: 5 273 261