Universidade Federal do Rio Grande do Sul

INF05501-B: Teoria da Computação N (2015/2)

Professor Tiaraju Asmuz Diverio

Rodrigo Ronconi Richter, 217445 e Lucas Bertolini Pizzo, 220484

Trabalho 09, Grupo 17

2: faça A := 3Q \* 5X(C) vá\_para End\_A

4: faça A := 3Q \* 5X(C) vá\_para End\_A

8: faça A := 3Q \* 5X(C) vá\_para End\_A

16: faça A := 3Q \* 5X(C) vá\_para End\_A 32: faça A := 3Q \* 5X(C) vá\_para End\_A

64: faça A := 3Q \* 5X(C) vá\_para End\_A

128: faça Y := X vá\_para fim

28125: faça X := 5 vá\_para 28126

28126: faça ad-C vá\_para 21827

28127: faça Q :=2 vá\_para End\_Q

9: faça X := 0 vá\_para 10

10: faça sub-C vá\_para 11

11: faça Q :=128 vá\_para End\_Q

1125: faça X := 3 vá\_para 1126

1126: faça sub-C vá\_para 1127

1127: faça Q :=128 vá\_para End\_Q

5525: faça X := 4 vá\_para 5626

5627: faça sub-C vá\_para 5628

5628: faça Q :=128 vá\_para End\_Q

45: faça X := 3 vá\_para 46

46: faça ad-C vá\_para 47

47: faça Q :=4 vá\_para End\_Q

225: faça X := 4 vá\_para 226

226: faça ad-C vá\_para 227

227: faça Q :=16 vá\_para End\_Q

405: faça X := 1 vá\_para 406

406: faça ad-C vá\_para 407

407: faça Q :=4 vá\_para End\_Q

2025: faça X := 2 vá\_para 2026

2026: faça ad-C vá\_para 2027

2027: faça Q :=4 vá\_para End\_Q

81: faça X := 0 vá\_para 82

82: faça sub-C vá\_para 83

83: faça Q :=8 vá\_para End\_Q

10125: faça X := 3 vá\_para 10126

10126: faça sub-C vá\_para 10127

10127: faça Q :=8 vá\_para End\_Q

50625: faça X := 4 vá\_para 50626

50626: faça sub-C vá\_para 50627

50627: faça Q :=8 vá\_para End\_Q

820125: faça X := 3 vá\_para 820126

820126: faça sub-C vá\_para 820827

820127: faça Q :=64 vá\_para End\_Q

1076168025: faça X := 4 vá\_para 1076168026

1076168026: faça ad-C vá\_para 1076168027

1076168027: faça Q :=16 vá\_para End\_Q

215233605: faça X := 2 vá\_para 215233606

215233606: faça ad-C vá\_para 215233607

215233607: faça Q :=16 vá\_para End\_Q

43046721: faça X := 0 vá\_para 43046722

43046722: faça sub-C vá\_para 43046723

43046723: faça Q :=32 vá\_para End\_Q

26904200625: faça X := 4 vá\_para 26904200626

26904200626: faça sub-C vá\_para 26904200627

26904200627: faça Q :=32 vá\_para End\_Q

5380840125: faça X := 3 vá\_para 5380840126

5380840126: faça sub-C vá\_para 5380840127

5380840127: faça Q :=32 vá\_para End\_Q

1158137618032: faça X := 4 vá\_para 1158137618033

1158137618033: faça sub-C vá\_para 1158137618034

1158137618034: faça Q :=64 vá\_para End\_Q

589509228886929…: faça X := 1 vá\_para 589509228886929…+1:

589509228886929…+1: faça sub-C vá\_para 589509228886929…+2:

589509228886929…+2: faça Q :=64 vá\_para End\_Q

294754614443464579…: faça X := 2 vá\_para 294754614443464579…+1

294754614443464579…+1: faça sub-C vá\_para 294754614443464579…+2

294754614443464579…+2: faça Q :=64 vá\_para End\_Q

147377307221732289644...: faça X := 3 vá\_para 147377307221732289644...+1

147377307221732289644...+1: faça ad-C vá\_para 147377307221732289644...+2

147377307221732289644...+2: faça Q :=2 vá\_para End\_Q

736886536108661448220054...: faça X := 3 vá\_para 736886536108661448220054...+1

736886536108661448220054...+1: faça ad-C vá\_para 736886536108661448220054...+2

736886536108661448220054...+2: faça Q :=2 vá\_para End\_Q

macro iterativo Q :=2 (Q)

ate zero-Q faca (sub-Q)

enquanto zero-Q faca (ad-Q;ad-Q)

fimmacro

macro iterativo Q :=4 (Q)

ate zero-Q faca (sub-Q)

enquanto zero-Q faca (ad-Q;ad-Q;ad-Q;ad-Q)

fimmacro

macro iterativo Q :=8 (Q)

ate zero-Q faca (sub-Q)

enquanto zero-Q faca (ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q)

fimmacro

macro iterativo Q :=16 (Q)

ate zero-Q faca (sub-Q)

enquanto zero-Q faca (ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q)

fimmacro

macro iterativo Q :=32 (Q)

ate zero-Q faca (sub-Q)

enquanto zero-Q faca (ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q)

fimmacro

macro iterativo Q :=64 (Q)

ate zero-Q faca (sub-Q)

enquanto zero-Q faca (ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q)

fimmacro

macro iterativo Q :=128 (Q)

ate zero-Q faca (sub-Q)

enquanto zero-Q faca (ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q; ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q;ad-Q)

fimmacro

macro iterativo copiar (Q, R, D)

ate zero-A faca (ad-C;ad-D;sub-A);

ate zero-D faca (ad-A;sub-D)

fimmacro

macro iterativo X := 0 (X, C)

ate zeroX(C) (X, C, E) faca subX(C) (X, C, E)

fimmacro

macro iterativo X := 1 (X, C)

ate zeroX(C) (X, C, E) faca subX(C) (X, C, E)

enquanto zeroX(C) (X, C, E) faca adX(C) (X, C, E)

fimmacro

macro iterativo X := 2 (X, C)

ate zeroX(C) (X, C, E) faca subX(C) (X, C, E)

enquanto zeroX(C) (X, C, E) faca ((adX(C) (X, C, E); (adX(C) (X, C, E))

fimmacro

macro iterativo X := 3 (X, C)

ate zeroX(C) (X, C, E) faca subX(C) (X, C, E)

enquanto zeroX(C) (X, C, E) faca ((adX(C) (X, C, E); (adX(C) (X, C, E); (adX(C) (X, C, E))

fimmacro

macro iterativo X := 4 (X, C)

ate zeroX(C) (X, C, E) faca subX(C) (X, C, E)

enquanto zeroX(C) (X, C, E) faca ((adX(C) (X, C, E); (adX(C) (X, C, E); (adX(C) (X, C, E); (adX(C) (X, C, E))

fimmacro

macro iterativo X := 5 (X, C)

ate zeroX(C) (X, C, E) faca subX(C) (X, C, E)

enquanto zeroX(C) (X, C, E) faca ((adX(C) (X, C, E); (adX(C) (X, C, E); (adX(C) (X, C, E); (adX(C) (X, C, E); (adX(C) (X, C, E))

fimmacro

macro monolitico End\_Q(Q, R)

{ vai para o rotulo de endereco Q }

R0: faça copiar(Q,R,D) vá\_para R1

R1: faça sub-R vá\_para R2

R2: faça sub-R vá\_para R3

R3: se zero-R então vá\_para 2 senão vá\_para R4

...

R135: se zero-R então vá\_para 128 senão vá\_para 0

fimmacro