jnt.scimark2.LU.factor(double[][], int[])

B0 [0, 0] . B1 0 Label ?

2 XIR: prologue()

0x0: e8 00 00 00 00 callq 0x5 //调用方法

0x5: 90 nop

0x6: 90 nop

0x7: 90 nop

0x8: 48 83 ec 38 sub $0x38,%rsp。 //开辟栈空间

0xc: 48 89 84 24 38 e0 ff mov %rax,-0x1fc8(%rsp)。

0x13: ff

B1 (SV)[0, 2] . B28 B27 dom B0 pred: B0 10 Label ?

12 XIR: rcx:i = arraylength(v0:a) input=(rdi:a) [bci:1, refmap(rsi:a, rdi:a)]

0x14: 48 8b 4f 10 mov 0x10(%rdi),%rcx //rcx是A.length=N

14 move rax:i = const[0|0x0]:i

0x18: b8 00 00 00 00 mov $0x0,%eax//eax存0

16 XIR: rdx:a = arrayload<Object>(v0:a, v4:i) input=(rdi:a, rax:i) temp=(rbx:i) [bci:5, refmap(rsi:a, rdi:a)]

0x1d: 48 63 5f 10 movslq 0x10(%rdi),%rbx//rbx是A.length

0x21: 3b c3 cmp %ebx,%eax//eax=0;比较是否超出数组范围

0x23: 0f 83 00 00 00 00 jae 0xc//若超出，则跳转异常

0x29: 48 8b 54 c7 18 mov 0x18(%rdi,%rax,8),%rdx//rdx是A[0]

18 XIR: rbx:i = arraylength(v3:a) input=(rdx:a) [bci:6, refmap(rdx:a, rsi:a, rdi:a)]

0x2e: 48 8b 5a 10 mov 0x10(%rdx),%rbx//rbx是A[0].length=M

20 Cmp (rbx:i, rcx:i)

0x32: 3b d9 cmp %ecx,%ebx//比较N和M大小

22 Branch > [B28]

0x34: 0f 8f 00 00 00 00 jg 0x6 //跳转块B28

B27 (V)[5, 6] . B29 dom B1 pred: B1 26 Label ?

28 move rdx:i = rbx:i

0x3a: 48 8b d3 mov %rbx,%rdx //rdx=minMN=M

30 Branch TRUE [B29]

0x3d: e9 00 00 00 00 jmpq 0x5 //跳转块29

B28 (V)[9, 9] . B29 dom B1 pred: B1 32 Label ?

34 move rdx:i = rcx:I

0x42: 48 8b d1 mov %rcx,%rdx // rdx=minMN=N

B29 (V)[10, 18] . B2 dom B1 pred: B28 B27 38 Label ?

40 move r8:i = const[0|0x0]:i

0x45: 41 b8 00 00 00 00 mov $0x0,%r8d //r8d=j=0

B2 (LHbV)[262, 266] . B24 B23 dom B29 pred: B17 B29 44 Label ?

0x4b: 90 nop

0x4c: 90 nop

0x4d: 90 nop

0x4e: 90 nop

0x4f: 90 nop

46 XIR: safepoint() temp=(r14:j) [bci:266, refmap(rsi:a, rdi:a)]

0x50: 4d 8b 36 mov (%r14),%r14

48 Cmp (r8:i, rdx:i)

0x53: 44 3b c2 cmp %edx,%r8d //比较j和minMN大小

50 Branch >= [B23]

0x56: 0f 8d 00 00 00 00 jge 0x6 //跳转到块23

B24 (V)[21, 43] . B3 dom B2 pred: B2 54 Label ?

56 XIR: rax:a = arrayload<Object>(v0:a, v8:i) input=(rdi:a, r8:i) temp=(r9:i) [bci:28, refmap(rsi:a, rdi:a)]

0x5c: 4c 63 4f 10 movslq 0x10(%rdi),%r9 //r9=A.length

0x60: 45 3b c1 cmp %r9d,%r8d //比较j和A.length大小，看A[j]有没有越界

0x63: 0f 83 00 00 00 00 jae 0xd //越界，异常跳转

0x69: 4a 8b 44 c7 18 mov 0x18(%rdi,%r8,8),%rax//rax=A[j]

58 XIR: xmm0:d = arrayload<double>(v9:a, v8:i) input=(rax:a, r8:i) temp=(r9:i) [bci:31, refmap(rax:a, rsi:a, rdi:a)]

0x6e: 4c 63 48 10 movslq 0x10(%rax),%r9 //r9=A[j].length

0x72: 45 3b c1 cmp %r9d,%r8 //比较j和A[j].length大小,看A[j][j]是否越界

0x75: 0f 83 00 00 00 00 jae 0xd //越界，异常跳转

0x7b: f2 42 0f 10 44 c0 18 movsd 0x18(%rax,%r8,8),%xmm0 //xmm0=A[j][j]

60 Abs xmm0:d = (xmm0:d, <illegal>)

0x82: 66 0f 54 05 00 00 00 andpd 0x0(%rip),%xmm0 # 0x8. //xmm0=t

0x89: 00

62 move rax:i = r8:i

0x8a: 49 8b c0 mov %r8,%rax. //rax=j

64 Add rax:i = (rax:i, const[1|0x1]:i)

0x8d: 83 c0 01 add $0x1,%eax //eax=j+1=i

66 move r9:i = rax:i

0x90: 4c 8b c8 mov %rax,%r9 //r9=i

70 move r10:i = r8:i

0x93: 4d 8b d0 mov %r8,%r10 //r10=jp

72 Branch TRUE [B3]

0x96: e9 00 00 00 00 jmpq 0x5 //跳转b3

B7 (bV)[46, 63] . B5 B4 dom B3 pred: B3 84 Label ?

0x9b: 90 nop

0x9c: 90 nop

0x9d: 90 nop

0x9e: 90 nop

0x9f: 90 nop

86 XIR: rax:a = arrayload<Object>(v0:a, v17:i) input=(rdi:a, r9:i) temp=(r12:i) [bci:49, refmap(rsi:a, rdi:a)]

0xa0: 4c 63 67 10 movslq 0x10(%rdi),%r12 //r12= A.length

0xa4: 45 3b cc cmp %r12d,%r9d //A[i]是否越界

0xa7: 0f 83 00 00 00 00 jae 0xd //越界跳转异常

0xad: 4a 8b 44 cf 18 mov 0x18(%rdi,%r9,8),%rax//rax=A[i]

88 XIR: xmm1:d = arrayload<double>(v18:a, v8:i) input=(rax:a, r8:i) temp=(r12:i) [bci:52, refmap(rax:a, rsi:a, rdi:a)]

0xb2: 4c 63 60 10 movslq 0x10(%rax),%r12 //r12=A[i].length

0xb6: 45 3b c4 cmp %r12d,%r8d //A[i][j]是否越界

0xb9: 0f 83 00 00 00 00 jae 0xd //越界跳转异常

0xbf: f2 42 0f 10 4c c0 18 movsd 0x18(%rax,%r8,8),%xmm1 //xmm1=A[i][j]

90 Abs xmm1:d = (xmm1:d, <illegal>)

0xc6: 66 0f 54 0d 00 00 00 andpd 0x0(%rip),%xmm1 # 0x8 //xmm1=ab

0xcd: 00

92 Cmp (xmm1:d, xmm0:d)

0xce: 66 0f 2e c8 ucomisd %xmm0,%xmm1 //比较t和ab

94 CondFloatBranch <= [B5]unordered: [B5]

0xd2: 0f 8a 00 00 00 00 jp 0x6 //if语句跳转分支

0xd8: 0f 86 00 00 00 00 jbe 0xc

B4 (V)[66, 72] . B5 dom B7 pred: B7 106 Label ?

108 move xmm0:d = xmm1:d

0xde: f2 0f 10 c1 movsd %xmm1,%xmm0 //t=ab

110 move r10:i = r9:i

0xe2: 4d 8b d1 mov %r9,%r10 //jp=i

B5 (V)[74, 74] . B3 dom B7 pred: B4 B7 114 Label ?

118 Add r9:i = (r9:i, const[1|0x1]:i)

0xe5: 41 83 c1 01 add $0x1,%r9d //i++

B3 (LHV)[77, 80] . B7 B6 dom B24 pred: B24 B5 74 Label ?

76 XIR: safepoint() temp=(r14:j) [bci:80, refmap(rsi:a, rdi:a)]

0xe9: 4d 8b 36 mov (%r14),%r14

78 Cmp (r9:i, rbx:i)

0xec: 44 3b cb cmp %ebx,%r9d //i和M比较大小

80 Branch < [B7]

0xef: 7c af jl 0xffffffb1 //跳转

B6 (V)[83, 98] . B9 B8 dom B3 pred: B3 128 Label ?

130 XIR: arraystore<int>(v1:a, v8:i, v15:i) input=(rsi:a, r8:i, r10:i) temp=(rax:i) [bci:88, refmap(rsi:a, rdi:a)]

0xf1: 48 63 46 10 movslq 0x10(%rsi),%rax //rax=pivot.length

0xf5: 44 3b c0 cmp %eax,%r8d //判断Pivot[j]是否越界

0xf8: 0f 83 00 00 00 00 jae 0xd //越界跳转异常

0xfe: 46 89 54 86 18 mov %r10d,0x18(%rsi,%r8,4) // Pivot[j]=jp

132 XIR: rax:a = arrayload<Object>(v0:a, v15:i) input=(rdi:a, r10:i) temp=(r12:i) [bci:92, refmap(rsi:a, rdi:a)]

0x103: 4c 63 67 10 movslq 0x10(%rdi),%r12

0x107: 45 3b d4 cmp %r12d,%r10d

0x10a: 0f 83 00 00 00 00 jae 0xd

0x110: 4a 8b 44 d7 18 mov 0x18(%rdi,%r10,8),%rax //rex=A[jp]

134 XIR: xmm1:d = arrayload<double>(v27:a, v8:i) input=(rax:a, r8:i) temp=(r12:i) [bci:95, refmap(rax:a, rsi:a, rdi:a)]

0x115: 4c 63 60 10 movslq 0x10(%rax),%r12

0x119: 45 3b c4 cmp %r12d,%r8d

0x11c: 0f 83 00 00 00 00 jae 0xd

0x122: f2 42 0f 10 4c c0 18 movsd 0x18(%rax,%r8,8),%xmm1 //xmm1=A[jp][j]

136 move xmm2:d = const[0.0|0x0]:d

0x129: 66 0f 57 d2 xorpd %xmm2,%xmm2 //xmm2=0

138 Cmp (xmm1:d, xmm2:d)

0x12d: 66 0f 2e ca ucomisd %xmm2,%xmm1. //判断A[jp][j]==0

140 CondFloatBranch == [B8]unordered: [B9] // if语句跳转分支

0x131: 0f 8a 00 00 00 00 jp 0x6

0x137: 0f 84 00 00 00 00 je 0xc

B9 (V)[103, 107] . B11 B10 dom B6 pred: B6 144 Label ?

146 Cmp (r10:i, r8:i)

0x13d: 45 3b d0 cmp %r8d,%r10d. //判断jp!=j

move stack:3:i = r9:i

0x140: 44 89 4c 24 18 mov %r9d,0x18(%rsp) //i入栈？？？

148 Branch == [B11]

0x145: 0f 84 00 00 00 00 je 0x6 //jp=j则跳转

B10 (V)[110, 129] . B11 dom B9 pred: B9 156 Label ?

158 XIR: rax:a = arrayload<Object>(v0:a, v8:i) input=(rdi:a, r8:i) temp=(r12:i) [bci:113, refmap(rsi:a, rdi:a)]

0x14b: 4c 63 67 10 movslq 0x10(%rdi),%r12

0x14f: 45 3b c4 cmp %r12d,%r8d

0x152: 0f 83 00 00 00 00 jae 0xd

0x158: 4a 8b 44 c7 18 mov 0x18(%rdi,%r8,8),%rax //rax=A[j]

160 XIR: r12:a = arrayload<Object>(v0:a, v15:i) input=(rdi:a, r10:i) temp=(r13:i) [bci:122, refmap(rax:a, rsi:a, rdi:a)]

0x15d: 4c 63 6f 10 movslq 0x10(%rdi),%r13

0x161: 45 3b d5 cmp %r13d,%r10d

0x164: 0f 83 00 00 00 00 jae 0xd

0x16a: 4e 8b 64 d7 18 mov 0x18(%rdi,%r10,8),%r12 //r12=A[jp]

162 XIR: arraystore<Object>(v0:a, v8:i, v34:a) input=(rdi:a, r8:i, r12:a) temp=(r13:i, r15:a, r9:a) [bci:123, refmap(rax:a, rsi:a, rdi:a, r12:a)]

0x16f: 4c 63 6f 10 movslq 0x10(%rdi),%r13

0x173: 45 3b c5 cmp %r13d,%r8d //判断j和A.length

0x176: 0f 83 00 00 00 00 jae 0xd //异常跳转

0x17c: 4d 33 db xor %r11,%r11 //r11=0

0x17f: 4d 3b e3 cmp %r11,%r12. //0和A[jp]大小

0x182: 0f 84 14 00 00 00 je 0x2d //异常跳转

0x188: 4c 8b 0f mov (%rdi),%r9 //r9

0x18b: 4d 8b 49 20 mov 0x20(%r9),%r9

0x18f: 4d 8b 3c 24 mov (%r12),%r15

0x193: 4d 3b cf cmp %r15,%r9

0x196: 0f 85 00 00 00 00 jne 0x2d. //异常跳转

0x19c: 4e 89 64 c7 18 mov %r12,0x18(%rdi,%r8,8) //A[j]=A[jp]

164 XIR: arraystore<Object>(v0:a, v15:i, v32:a) input=(rdi:a, r10:i, rax:a) temp=(r9:i, r12:a, r13:a) [bci:129, refmap(rax:a, rsi:a, rdi:a)]

0x1a1: 4c 63 4f 10 movslq 0x10(%rdi),%r9

0x1a5: 45 3b d1 cmp %r9d,%r10d //判断jp和A.length

0x1a8: 0f 83 00 00 00 00 jae 0xd //异常跳转

0x1ae: 4d 33 db xor %r11,%r11

0x1b1: 49 3b c3 cmp %r11,%rax //rax=tA[]

0x1b4: 0f 84 13 00 00 00 je 0x2c

0x1ba: 4c 8b 2f mov (%rdi),%r13

0x1bd: 4d 8b 6d 20 mov 0x20(%r13),%r13

0x1c1: 4c 8b 20 mov (%rax),%r12.

0x1c4: 4d 3b ec cmp %r12,%r13

0x1c7: 0f 85 00 00 00 00 jne 0x2c

0x1cd: 4a 89 44 d7 18 mov %rax,0x18(%rdi,%r10,8) //A[jp]=tA

B11 (V)[130, 135] . B13 B12 dom B9 pred: B10 B9 168 Label ?

170 move rax:i = rbx:i

0x1d2: 48 8b c3 mov %rbx,%rax //rax=M

172 Sub rax:i = (rax:i, const[1|0x1]:i)

0x1d5: 83 e8 01 sub $0x1,%eax //eax=M-1

174 Cmp (r8:i, rax:i)

0x1d8: 44 3b c0 cmp %eax,%r8d //判断j<M-1

176 Branch >= [B13]

0x1db: 0f 8d 00 00 00 00 jge 0x6 //跳转

B12 (V)[138, 155] . B14 dom B11 pred: B11 184 Label ?

186 XIR: rax:a = arrayload<Object>(v0:a, v8:i) input=(rdi:a, r8:i) temp=(r9:i) [bci:142, refmap(rsi:a, rdi:a)]

0x1e1: 4c 63 4f 10 movslq 0x10(%rdi),%r9

0x1e5: 45 3b c1 cmp %r9d,%r8d

0x1e8: 0f 83 00 00 00 00 jae 0xd

0x1ee: 4a 8b 44 c7 18 mov 0x18(%rdi,%r8,8),%rax //rax=A[j]

188 XIR: xmm1:d = arrayload<double>(v43:a, v8:i) input=(rax:a, r8:i) temp=(r9:i) [bci:145, refmap(rax:a, rsi:a, rdi:a)]

0x1f3: 4c 63 48 10 movslq 0x10(%rax),%r9

0x1f7: 45 3b c1 cmp %r9d,%r8d

0x1fa: 0f 83 00 00 00 00 jae 0xd

0x200: f2 42 0f 10 4c c0 18 movsd 0x18(%rax,%r8,8),%xmm1. // xmm1=A[j][j]

190 move xmm2:d = const[1.0|0x3ff0000000000000]:d

0x207: 49 bb 00 00 00 00 00 movabs $0x3ff0000000000000,%r11. //r11=1.0

0x20e: 00 f0 3f

0x211: 66 49 0f 6e d3 movq %r11,%xmm2 //xmm2=1.0

194 Div xmm2:d = (xmm2:d, xmm1:d)

0x216: f2 0f 5e d1 divsd %xmm1,%xmm2 //xmm2=recp

196 move rax:i = r8:i

0x21a: 49 8b c0 mov %r8,%rax //rax=j

198 Add rax:i = (rax:i, const[1|0x1]:i)

0x21d: 83 c0 01 add $0x1,%eax //eax=j+1=k

202 Branch TRUE [B14]

0x220: e9 00 00 00 00 jmpq 0x5 //跳转

B15 (bV)[158, 170] . B14 dom B14 pred: B14 214 Label ?

0x225: 90 nop

0x226: 90 nop

0x227: 90 nop

216 XIR: r9:a = arrayload<Object>(v0:a, v50:i) input=(rdi:a, rax:i) temp=(r12:i) [bci:161, refmap(rsi:a, rdi:a)]

0x228: 4c 63 67 10 movslq 0x10(%rdi),%r12 //r12=A[k]

0x22c: 41 3b c4 cmp %r12d,%eax

0x22f: 0f 83 00 00 00 00 jae 0xd

0x235: 4c 8b 4c c7 18 mov 0x18(%rdi,%rax,8),%r9 //r9=A[k]

218 XIR: xmm1:d = arrayload<double>(v51:a, v8:i) input=(r9:a, r8:i) temp=(r12:i) [bci:165, refmap(rsi:a, rdi:a, r9:a)]

0x23a: 4d 63 61 10 movslq 0x10(%r9),%r12 //r12=A[k][j]

0x23e: 45 3b c4 cmp %r12d,%r8d

0x241: 0f 83 00 00 00 00 jae 0xd

0x247: f2 43 0f 10 4c c1 18 movsd 0x18(%r9,%r8,8),%xmm1 // xmm1=A[k][j]

222 Mul xmm1:d = (xmm1:d, xmm2:d)

0x24e: f2 0f 59 ca mulsd %xmm2,%xmm1 //xmm1=A[k][j]\*recp

224 XIR: arraystore<double>(v51:a, v8:i, v55:d) input=(r9:a, r8:i, xmm1:d) temp=(r12:i) [bci:169, refmap(rsi:a, rdi:a, r9:a)]

0x252: 4d 63 61 10 movslq 0x10(%r9),%r12 //r12=A[k].length

0x256: 45 3b c4 cmp %r12d,%r8d //判断A[k][j]是否越界

0x259: 0f 83 00 00 00 00 jae 0xd

0x25f: f2 43 0f 11 4c c1 18 movsd %xmm1,0x18(%r9,%r8,8)//A[k][j]=A[k][j]\*recp

228 Add rax:i = (rax:i, const[1|0x1]:i)

0x266: 83 c0 01 add $0x1,%eax //k++

B14 (LHV)[173, 176] . B15 B13 dom B12 pred: B12 B15 204 Label ?

206 XIR: safepoint() temp=(r14:j) [bci:176, refmap(rsi:a, rdi:a)]

0x269: 4d 8b 36 mov (%r14),%r14

208 Cmp (rax:i, rbx:i)

0x26c: 3b c3 cmp %ebx,%eax //比较k和M大小

210 Branch < [B15]

0x26e: 7c b8 jl 0xffffffba

B13 (V)[179, 185] . B17 B16 dom B11 pred: B11 B14 238 Label ?

240 move rax:i = rdx:i

0x270: 48 8b c2 mov %rdx,%rax //rax=minMN

242 Sub rax:i = (rax:i, const[1|0x1]:i)

0x273: 83 e8 01 sub $0x1,%eax //eax=minMN-1

244 Cmp (r8:i, rax:i)

0x276: 44 3b c0 cmp %eax,%r8d //判断j<minMN

246 Branch >= [B17]

0x279: 0f 8d 00 00 00 00 jge 0x6

B16 (V)[188, 194] . B18 dom B13 pred: B13 254 Label ?

256 move rax:i = r8:i

0x27f: 49 8b c0 mov %r8,%rax

258 Add rax:i = (rax:i, const[1|0x1]:i)

0x282: 83 c0 01 add $0x1,%eax

262 Branch TRUE [B18]

0x285: e9 00 00 00 00 jmpq 0x5

B22 (bV)[197, 222] . B19 dom B18 pred: B18 274 Label ?

0x28a: 90 nop

0x28b: 90 nop

0x28c: 90 nop

0x28d: 90 nop

0x28e: 90 nop

0x28f: 90 nop

276 XIR: r9:a = arrayload<Object>(v0:a, v60:i) input=(rdi:a, rax:i) temp=(r12:i) [bci:200, refmap(rsi:a, rdi:a)]

0x290: 4c 63 67 10 movslq 0x10(%rdi),%r12

0x294: 41 3b c4 cmp %r12d,%eax

0x297: 0f 83 00 00 00 00 jae 0xd

0x29d: 4c 8b 4c c7 18 mov 0x18(%rdi,%rax,8),%r9

278 XIR: r12:a = arrayload<Object>(v0:a, v8:i) input=(rdi:a, r8:i) temp=(r13:i) [bci:206, refmap(rsi:a, rdi:a, r9:a)]

0x2a2: 4c 63 6f 10 movslq 0x10(%rdi),%r13

0x2a6: 45 3b c5 cmp %r13d,%r8d

0x2a9: 0f 83 00 00 00 00 jae 0xd

0x2af: 4e 8b 64 c7 18 mov 0x18(%rdi,%r8,8),%r12

280 XIR: xmm1:d = arrayload<double>(v61:a, v8:i) input=(r9:a, r8:i) temp=(r13:i) [bci:213, refmap(rsi:a, rdi:a, r9:a, r12:a)]

0x2b4: 4d 63 69 10 movslq 0x10(%r9),%r13

0x2b8: 45 3b c5 cmp %r13d,%r8d

0x2bb: 0f 83 00 00 00 00 jae 0xd

0x2c1: f2 43 0f 10 4c c1 18 movsd 0x18(%r9,%r8,8),%xmm1

282 move r13:i = r8:i

0x2c8: 4d 8b e8 mov %r8,%r13

284 Add r13:i = (r13:i, const[1|0x1]:i)

0x2cb: 41 83 c5 01 add $0x1,%r13d

288 Branch TRUE [B19]

0x2cf: e9 00 00 00 00 jmpq 0x5

B21 (bV)[225, 241] . B19 dom B19 pred: B19 300 Label ?

0x2d4: 90 nop

0x2d5: 90 nop

0x2d6: 90 nop

0x2d7: 90 nop

302 XIR: xmm2:d = arrayload<double>(v61:a, v68:i) input=(r9:a, r13:i) temp=(r15:i) [bci:230, refmap(rsi:a, rdi:a, r9:a, r12:a)]

0x2d8: 4d 63 79 10 movslq 0x10(%r9),%r15

0x2dc: 45 3b ef cmp %r15d,%r13d

0x2df: 0f 83 00 00 00 00 jae 0xd

0x2e5: f2 43 0f 10 54 e9 18 movsd 0x18(%r9,%r13,8),%xmm2

304 XIR: xmm3:d = arrayload<double>(v63:a, v68:i) input=(r12:a, r13:i) temp=(r15:i) [bci:237, refmap(rsi:a, rdi:a, r9:a, r12:a)]

0x2ec: 4d 63 7c 24 10 movslq 0x10(%r12),%r15

0x2f1: 45 3b ef cmp %r15d,%r13d

0x2f4: 0f 83 00 00 00 00 jae 0xe

0x2fa: f2 43 0f 10 5c ec 18 movsd 0x18(%r12,%r13,8),%xmm3

308 Mul xmm3:d = (xmm3:d, xmm1:d)

0x301: f2 0f 59 d9 mulsd %xmm1,%xmm3

312 Sub xmm2:d = (xmm2:d, xmm3:d)

0x305: f2 0f 5c d3 subsd %xmm3,%xmm2

314 XIR: arraystore<double>(v61:a, v68:i, v74:d) input=(r9:a, r13:i, xmm2:d) temp=(r15:i) [bci:240, refmap(rsi:a, rdi:a, r9:a, r12:a)]

0x309: 4d 63 79 10 movslq 0x10(%r9),%r15

0x30d: 45 3b ef cmp %r15d,%r13d

0x310: 0f 83 00 00 00 00 jae 0xd

0x316: f2 43 0f 11 54 e9 18 movsd %xmm2,0x18(%r9,%r13,8)

318 Add r13:i = (r13:i, const[1|0x1]:i)

0x31d: 41 83 c5 01 add $0x1,%r13d

B19 (LHV)[244, 247] . B21 B20 dom B22 pred: B22 B21 290 Label ?

292 XIR: safepoint() temp=(r14:j) [bci:247, refmap(rsi:a, rdi:a, r9:a, r12:a)]

0x321: 4d 8b 36 mov (%r14),%r14

294 Cmp (r13:i, rcx:i)

0x324: 44 3b e9 cmp %ecx,%r13d

296 Branch < [B21]

0x327: 7c af jl 0xffffffb1

B20 (V)[250, 250] . B18 dom B19 pred: B19 324 Label ?

328 Add rax:i = (rax:i, const[1|0x1]:i)

0x329: 83 c0 01 add $0x1,%eax

B18 (LHV)[253, 256] . B22 B17 dom B16 pred: B16 B20 264 Label ?

266 XIR: safepoint() temp=(r14:j) [bci:256, refmap(rsi:a, rdi:a)]

0x32c: 4d 8b 36 mov (%r14),%r14

268 Cmp (rax:i, rbx:i)

0x32f: 3b c3 cmp %ebx,%eax

270 Branch < [B22]

0x331: 0f 8c 59 ff ff ff jl 0xffffff5f

B17 (V)[259, 259] . B2 dom B13 pred: B13 B18 338 Label ?

342 Add r8:i = (r8:i, const[1|0x1]:i)

0x337: 41 83 c0 01 add $0x1,%r8d

346 Branch TRUE [B2]

0x33b: e9 10 fd ff ff jmpq 0xfffffd15

B8 (V)[101, 102] dom B6 pred: B6 348 Label ?

350 move rax:i = const[1|0x1]:i

0x340: b8 01 00 00 00 mov $0x1,%eax

352 XIR: epilogue() method=jnt.scimark2.LU.factor(double[][], int[]) [bci:102, refmap(rsi:a, rdi:a)]

0x345: 48 83 c4 38 add $0x38,%rsp

354 Return rax:i

0x349: c3 retq

B23 (V)[269, 270] dom B2 pred: B2 356 Label ?

358 move rax:i = const[0|0x0]:i

0x34a: b8 00 00 00 00 mov $0x0,%eax

360 XIR: epilogue() method=jnt.scimark2.LU.factor(double[][], int[]) [bci:270, refmap(rsi:a, rdi:a)]

0x34f: 48 83 c4 38 add $0x38,%rsp

362 Return rax:i

0x353: c3 retq