Assignment 1117

Problem 1

Liveness Analysis

```
u: RAX = read_int()
def(u) = RAX
liveness(u) = {}
```

```
u: #0 = RAX
def(u) = #0
use(u) = RAX
liveness(u) = {RAX}
```

```
u: #1 = 1
def(u) = #1
liveness(u) = {#0}
```

```
u: #2 = 0
def(u) = #2
liveness(u) = {#0, #1}
```

```
u: RAX = read_int()
def(u) = RAX
liveness(u) = {#0, #1, #2}
```

```
u: #5 = RAX
def(u) = #5
use(u) = RAX
liveness(u) = {#0, #1, #2, RAX}
```

```
u: RAX = read_int()
def(u) = RAX
liveness(u) = {#0, #1, #2, #5}
```

```
u: #6 = RAX
def(u) = #6
use(u) = RAX
liveness(u) = {#0, #1, #2, #5, RAX}
```

```
u: RAX = read_int()
def(u) = RAX
liveness(u) = {#0, #1, #2, #6, #5}
```

```
u: #7 = RAX
def(u) = #7
use(u) = RAX
liveness(u) = {#0, #1, #2, #6, #5, RAX}
```

```
u: RAX = read_int()
def(u) = RAX
liveness(u) = {#0, #1, #2, #6, #5, #7}
```

```
u: #8 = RAX
def(u) = #8
use(u) = RAX
liveness(u) = {#0, #1, #2, #6, #5, #7, RAX}
```

```
u: jmp 1
liveness(u) = {#0, #1, #2, #6, #5, #7, #8}
```

```
u: if (GT(#0, 0)) then jmp 2 else jmp 3
use(u) = #0
liveness(u) = {#0, #1, #2,  #6, #5, #7, #8}
```

```
u: #9 = MUL(#1, #5)
def(u) = #9
use(u) = #1, #5
liveness(u) = {#0, #1, #2, #6, #5, #7, #8}
```

```
u: #10 = MUL(#2, #7)
def(u) = #10
use(u) = #2, #7
liveness(u) = {#0, #1, #2, #6, #5, #7, #8, #9}
```

```
u: #3 = PLUS(#9, #10)
def(u) = #3
use(u) = #9, #10
liveness(u) = {#0, #1, #2, #6, #5, #7, #8, #9, #10}
```

```
u: #11 = MUL(#1, #6)
def(u) = #11,
use(u) = #1, #6
liveness(u) = {#0, #1, #2, #3, #6, #5, #7, #8}
```

```
u: #12 = MUL(#2, #8)
def(u) = #12
use(u) = #2, #8
liveness(u) = {#0,#2, #3, #6, #5, #7, #8, #11}
```

```
u: #4 = PLUS(#11, #12)
def(u) = #4
use(u) = #11, #12
liveness(u) =  {#0, #3, #6, #5, #7, #8, #11, #12}
```

```
u: #1 = #3
def(u) = #1
use(u) = #3
liveness(u) = {#0, #4, #3, #6, #5, #7, #8}
```

```
u: #2 = #4
def(u) = #2
use(u) = #4
liveness(u) = {#0, #4,  #1, #6, #5, #7, #8}
```

```
u: #0 = MINUS(#0, 1)
def(u) = #0
use(u) = #0
liveness(u) = {#0, #1, #2,  #6, #5, #7, #8}
```

```
u: jmp 1
liveness(u)={#0, #1, #2,  #6, #5, #7, #8}
```

Inference Graph

- #0: {#1, #2, #3, #4, #5, #6, #7, #8, #9, #10, #11, #12, RAX} [13]
- #1: {#0, #2, #3, #4, #5, #6, #7, #8, #9, #10, RAX}[11]
- #2: {#0, #1, #3, #5, #6, #7, #8, #9, #10, #11, RAX}[11]
- #3: {#0, #1, #2, #4, #5, #6, #7, #8, #11, #12}[10]
- #4: {#0, #1, #3, #5, #6, #7, #8}[7]
- #5: {#0, #1, #2, #3, #4, #6, #7, #8, #9, #10, #11, #12, RAX} [13]
- #6: {#0, #1, #2, #3, #4, #5, #7, #8, #9, #10, #11, #12, RAX} [13]
- #7: {#0, #1, #2, #3, #4, #5, #6, #8, #9, #10, #11, #12, RAX} [13]
- #8: {#0, #1, #2, #3, #4, #5, #6, #7, #9, #10, #11, #12}[12]
- #9: {#0, #1, #2, #5, #6, #7, #8, #10}[8]
- #10: {#0, #1, #2, #5, #6, #7, #8, #9}[8]
- #11: {#0, #2, #3, #5, #6, #7, #8, #12}[8]
- #12: {#0, #3, #5, #6, #7, #8, #11}[7]
- #RAX: {#0, #1, #2, #5, #6, #7}[6]

K = 9, K - 1 = 8

与move有关的点: {0, 1, 2, 3, 4, 5, 6, 7, 8, RAX}, move 涉及: `#2 = #4`, `#1 = #3`, #5,6,7,8 与 RAX

1. Simplify. #9, #10, #11, #12 均满足条件, 删除后图简化为:
    - #0: {#1, #2, #3, #4, #5, #6, #7, #8, RAX} [9]
    - #1: {#0, #2, #3, #4, #5, #6, #7, #8, RAX}[9]
    - #2: {#0, #1, #3, #5, #6, #7, #8, RAX}[8]
    - #3: {#0, #1, #2, #4, #5, #6, #7, #8}[8]
    - #4: {#0, #1, #3, #5, #6, #7, #8}[7]
    - #5: {#0, #1, #2, #3, #4, #6, #7, #8, RAX} [9]
    - #6: {#0, #1, #2, #3, #4, #5, #7, #8, RAX} [9]
    - #7: {#0, #1, #2, #3, #4, #5, #6, #8, RAX} [9]
    - #8: {#0, #1, #2, #3, #4, #5, #6, #7}[8]
    - RAX: {#0, #1, #2, #5, #6, #7}[6]
2. Coalesce
    - #2的邻居中度数大于等于K = 9的有 #0, #1, #5, #6, #7
    - #4的邻居中度数大于等于K = 9的有 #0, #1, #5, #6, #7
    - #1的邻居中度数大于等于K = 9的有 #0, #5, #6, #7
    - #3的邻居中度数大于等于K = 9的有 #0, #1, #5, #6, #7
    - RAX的邻居中度数大于等于K = 9有 #0, #1, #5, #6, #7

    继续分析，最终满足要求的move指令有RAX与#8 ； #2与#4；进行保守合并，则Inference Graph 变为:
    - #0: {#1, #2, #3, #5, #6, #7, RAX} [7]
    - #1: {#0, #2, #3, #5, #6, #7, RAX}[7]
    - #2: {#0, #1, #3, #5, #6, #7, RAX}[7]
    - #3: {#0, #1, #2, #5, #6, #7, RAX}[7]
    - #5: {#0, #1, #2, #3, #6, #7, RAX} [7]
    - #6: {#0, #1, #2, #3, #5, #7, RAX} [7]
    - #7: {#0, #1, #2, #3, #5, #6, RAX} [7]
    - RAX: {#0, #1, #2, #5, #6, #7}[6]
3. Simplify #0, #1,#2,#3, #5,#6,#7, RAX
    - #9: {}[0]

分配寄存器:

```
x0: RAX, #8
x1: #7
x2: #6
x3: #5
x4: #3, #10
x5: #2, #4
x6: #1, #11
x7: #0
x8: #9, #12
```

Problem 2

分配寄存器前的步骤Problem 1中已完成。

- #0: {#1, #2, #3, #4, #5, #6, #7, #8, #9, #10, #11, #12, RAX} [13]
- #1: {#0, #2, #3, #4, #5, #6, #7, #8, #9, #10, RAX}[11]
- #2: {#0, #1, #3, #5, #6, #7, #8, #9, #10, #11, RAX}[11]
- #3: {#0, #1, #2, #4, #5, #6, #7, #8, #11, #12}[10]
- #4: {#0, #1, #3, #5, #6, #7, #8}[7]
- #5: {#0, #1, #2, #3, #4, #6, #7, #8, #9, #10, #11, #12, RAX} [13]
- #6: {#0, #1, #2, #3, #4, #5, #7, #8, #9, #10, #11, #12, RAX} [13]
- #7: {#0, #1, #2, #3, #4, #5, #6, #8, #9, #10, #11, #12, RAX} [13]
- #8: {#0, #1, #2, #3, #4, #5, #6, #7, #9, #10, #11, #12}[12]
- #9: {#0, #1, #2, #5, #6, #7, #8, #10}[8]
- #10: {#0, #1, #2, #5, #6, #7, #8, #9}[8]
- #11: {#0, #2, #3, #5, #6, #7, #8, #12}[8]
- #12: {#0, #3, #5, #6, #7, #8, #11}[7]
- #RAX: {#0, #1, #2, #5, #6, #7}[6]

K = 7, K - 1 = 6

与move有关的点: {0, 1, 2, 3, 4, 5, 6, 7, 8, RAX}, move 涉及: `#2 = #4`, `#1 = #3`, #0, 5,6,7,8 与 RAX

1. simplify，无法simplify
2. coalesce，#8和RAX合并。
    - #0: {#1, #2, #3, #4, #5, #6, #7, #8, #9, #10, #11, #12} [12]
    - #1: {#0, #2, #3, #4, #5, #6, #7, #8, #9, #10}[10]
    - #2: {#0, #1, #3, #5, #6, #7, #8, #9, #10, #11}[10]
    - #3: {#0, #1, #2, #4, #5, #6, #7, #8, #11, #12}[10]

    - #4: {#0, #1, #3, #5, #6, #7, #8}[7]
    - #5: {#0, #1, #2, #3, #4, #6, #7, #8, #9, #10, #11, #12} [12]
    - #6: {#0, #1, #2, #3, #4, #5, #7, #8, #9, #10, #11, #12} [12]
    - #7: {#0, #1, #2, #3, #4, #5, #6, #8, #9, #10, #11, #12} [12]
    - #8: {#0, #1, #2, #3, #4, #5, #6, #7, #9, #10, #11, #12}[12]
    - #9: {#0, #1, #2, #5, #6, #7, #8, #10}[8]
    - #10: {#0, #1, #2, #5, #6, #7, #8, #9}[8]
    - #11: {#0, #2, #3, #5, #6, #7, #8, #12}[8]
    - #12: {#0, #3, #5, #6, #7, #8, #11}[7]
3. simplify，无法simplify
4. coalesce，#2和#4合并，
    - #0: {#1, #2, #3, #5, #6, #7, #8, #9, #10, #11, #12} [11]
    - #1: {#0, #2, #3, #5, #6, #7, #8, #9, #10}[9]
    - #2: {#0, #1, #3, #5, #6, #7, #8, #9, #10, #11}[10]
    - #3: {#0, #1, #2, #5, #6, #7, #8, #11, #12}[9]
    - #5: {#0, #1, #2, #3, #6, #7, #8, #9, #10, #11, #12} [11]
    - #6: {#0, #1, #2, #3, #5, #7, #8, #9, #10, #11, #12} [11]
    - #7: {#0, #1, #2, #3, #5, #6, #8, #9, #10, #11, #12} [11]
    - #8: {#0, #1, #2, #3, #5, #6, #7, #9, #10, #11, #12}[11]
    - #9: {#0, #1, #2, #5, #6, #7, #8, #10}[8]
    - #10: {#0, #1, #2, #5, #6, #7, #8, #9}[8]
    - #11: {#0, #2, #3, #5, #6, #7, #8, #12}[8]
    - #12: {#0, #3, #5, #6, #7, #8, #11}[7]
5. spill #0
    - #1: {#2, #3, #5, #6, #7, #8, #9, #10}[8]
    - #2: {#1, #3, #5, #6, #7, #8, #9, #10, #11}[9]
    - #3: {#1, #2, #5, #6, #7, #8, #11, #12}[8]
    - #5: {#1, #2, #3, #6, #7, #8, #9, #10, #11, #12} [10]
    - #6: {#1, #2, #3, #5, #7, #8, #9, #10, #11, #12} [10]
    - #7: {#1, #2, #3, #5, #6, #8, #9, #10, #11, #12} [10]
    - #8: {#1, #2, #3, #5, #6, #7, #9, #10, #11, #12}[10]
    - #9: {#1, #2, #5, #6, #7, #8, #10}[7]

- #10: {#1, #2, #5, #6, #7, #8, #9}[7]
- #11: {#2, #3, #5, #6, #7, #8, #12}[7]
- #12: {#3, #5, #6, #7, #8, #11}[6]

6. simplify #12
   - #1: {#2, #3, #5, #6, #7, #8, #9, #10}[8]
   - #2: {#1, #3, #5, #6, #7, #8, #9, #10, #11}[9]
   - #5: {#1, #2, #3, #6, #7, #8, #9, #10, #11} [10]
   - #6: {#1, #2, #3, #5, #7, #8, #9, #10, #11} [9]
   - #7: {#1, #2, #3, #5, #6, #8, #9, #10, #11} [9]
   - #8: {#1, #2, #3, #5, #6, #7, #9, #10, #11}[9]
   - #9: {#1, #2, #5, #6, #7, #8, #10}[7]
   - #10: {#1, #2, #5, #6, #7, #8, #9}[7]
7. simplify #11, #3
   - #1: {#2, #5, #6, #7, #8, #9, #10}[7]
   - #2: {#1, #5, #6, #7, #8, #9, #10}[7]
   - #5: {#1, #2, #6, #7, #8, #9, #10} [8]
   - #6: {#1, #2, #5, #7, #8, #9, #10} [7]
   - #7: {#1, #2, #5, #6, #8, #9, #10} [7]
   - #8: {#1, #2, #5, #6, #7, #9, #10}[7]
   - #9: {#1, #2, #5, #6, #7, #8, #10}[7]
   - #10: {#1, #2, #5, #6, #7, #8, #9}[7]
8. Spill #1
9. Simplify #9, #10
10. Simplify #5, #6, #7, #14
11. Simplify #13

寄存器分配：

```
x0: #2, #4, #12
x1: RAX, #8
x2: #7
x3: #6
x4: #5
x5: #10, #3
x6: #9, #11
MEM(spilled): #1, #0
```

出现了真spill，所以要startover。

```
u: RAX = read_int()
def(u) = RAX
liveness(u) = {}
```

```
u: #0 = RAX
def(u) = #0
use(u) = RAX
liveness(u) = {RAX}
```

```
u: *(%rbp - 16)= #0
use(u) = #0
liveness(u) = {#0}
```

```
u: #1 = 1
def(u) = #1
liveness(u) = {}
```

```
u: *(%rbp - 32) = #1
use(u) = #1
liveness(u) = {#1}
```

```
u: #2 = 0
def(u) = #2
liveness(u) = {}
```

```
u: RAX = read_int()
def(u) = RAX
liveness(u) = {#2}
```

```
u: #5 = RAX
def(u) = #5
use(u) = RAX
liveness(u) = {#2, RAX}
```

```
u: RAX = read_int()
def(u) = RAX
liveness(u) = {#5, #2}
```

```
u: #6 = RAX
def(u) = #6
use(u) = RAX
liveness(u) = {#5, #2, RAX}
```

```
u: RAX = read_int()
def(u) = RAX
liveness(u) = {#5, #6, #2}
```

```
u: #7 = RAX
def(u) = #7
use(u) = RAX
liveness(u) = {#5, #6, #2, RAX}
```

```
u: RAX = read_int()
def(u) = RAX
liveness(u) = {#5, #6, #2, #7}
```

```
u: #8 = RAX
def(u) = #8
use(u) = RAX
liveness(u) = {#5, #6, #2, #7, RAX}
```

```
u: jmp 1
liveness(u) = {#5, #6, #2, #8, #7}
```

```
u: #0 = *(%rbp - 16)
def(u) = #0
liveness(u) = {#5, #6, #2, #8, #7}
```

```
u: if (GT(#0, 0)) then jmp 2 else jmp 3
use(u) = #0
liveness(u) = {#0, #5, #6, #2, #8, #7}
```

```
u: #1 = *(%rbp - 32)
def(u) = #1
liveness(u) = {#5, #6, #2, #8, #7}
```

```
u: #9 = MUL(#1, #5)
def(u) = #9
use(u) = #1, #5
liveness(u) = {#1, #5, #6, #2, #8, #7}
```

```
u: #10 = MUL(#2, #7)
def(u) = #10
use(u) = #2, #7
liveness(u) = {#6, #2, #8, #9, #7,  #5}
```

```
u: #3 = PLUS(#9, #10)
def(u) = #3
use(u) = #9, #10
liveness(u) = {#9, #10, #6, #2, #8,  #5,  #7}
```

```
u: #1 = *(%rbp - 32)
def(u) = #1
liveness(u) = {#3,#6, #2, #8,  #5,  #7}
```

```
u: #11 = MUL(#1, #6)
def(u) = #11
use(u) = #1, #6
liveness(u) = {#3, #1, #6, #2, #8, #5,  #7}
```

```
u: #12 = MUL(#2, #8)
def(u) = #12
use(u) = #2, #8
liveness(u) = {#3, #11, #2,  #5, #6,  #8, #7}
```

```
u: #4 = PLUS(#11, #12)
def(u) = #4
use(u) = #11, #12
liveness(u) = {#3, #11, #12, #5, #6,  #8, #7}
```

```
u: #1 = #3
def(u) = #1
use(u) = #3
liveness(u) = {#3, #4,  #5, #6,  #8, #7}
```

```
u: *(%rbp - 32) = #1
use(u) = #1
liveness(u) = {#4, #1, #5, #6,  #8, #7}
```

```
u: #2 = #4
def(u) = #2
use(u) = #4
liveness(u) = {#4, #5, #6,  #8, #7}
```

```
u: #0 = *(%rbp - 16)
def(u) = #0
liveness(u) = {#5, #6, #2, #8, #7}
```

```
u: #0 = MINUS(#0, 1)
def(u) = #0
use(u) = #0
liveness(u) = {#0, #5, #6, #2, #8, #7}
```

```
u: *(%rbp - 16) = #0
use(u) = #0
liveness(u) = {#0, #5, #6, #2, #8, #7}
```

```
u: jmp 1
liveness(u)= {#0, #5, #6, #2, #8, #7}
```

Inference Graph

- #0, Set: {'#6', '#2', '#8', '#5', '#7'}, Length: 5
- #1, Set: {'#6', '#2', '#8', '#5', '#3', '#4', '#7'}, Length: 7
- #2, Set: {'#6', '#11', 'RAX', '#0', '#8', '#9', '#10', '#5', '#3', '#1', '#7'}, Length: 11
- #3, Set: {'#6', '#11', '#2', '#8', '#5', '#4', '#1', '#7', '#12'}, Length: 9
- #4, Set: {'#6', '#8', '#5', '#1', '#7', '#3'}, Length: 6
- #5, Set: {'#6', '#11', 'RAX', '#2', '#0', '#8', '#9', '#10', '#3', '#4', '#1', '#7', '#12'}, Length: 13
- #6, Set: {'#11', 'RAX', '#2', '#0', '#8', '#9', '#10', '#5', '#3', '#4', '#1', '#7', '#12'}, Length: 13
- #7, Set: {'#6', '#11', 'RAX', '#2', '#0', '#8', '#9', '#10', '#5', '#3', '#4', '#1', '#12'}, Length: 13
- #8, Set: {'#6', '#11', '#2', '#0', '#9', '#10', '#5', '#3', '#4', '#1', '#7', '#12'}, Length: 12
- #9, Set: {'#6', '#2', '#8', '#10', '#5', '#7'}, Length: 6
- #10, Set: {'#6', '#2', '#8', '#5', '#9', '#7'}, Length: 6
- #11, Set: {'#6', '#2', '#8', '#5', '#7', '#3', '#12'}, Length: 7
- #12, Set: {'#6', '#11', '#8', '#5', '#7', '#3'}, Length: 6
- RAX, Set: {'#6', '#2', '#5', '#7'}, Length: 4

K = 7, K - 1 = 6

与move有关的点: {0, 1, 2, 3, 4, 5, 6, 7, 8, RAX}, move 涉及: `#2 = #4`, `#1 = #3`, #0, 5,6,7,8 与 RAX。 而#1和#3live区域重叠，不考虑。

1. Simplify. #9, #10, #12 均可以消去
    - #0, Set: {'#6', '#2', '#8', '#5', '#7'}, Length: 5
    - #1, Set: {'#6', '#2', '#8', '#5', '#3', '#4', '#7'}, Length: 7
    - #2, Set: {'#6', '#11', 'RAX', '#0', '#8', '#5', '#3', '#1', '#7'}, Length: 9
    - #3, Set: {'#6', '#11', '#2', '#8', '#5', '#4', '#1', '#7'}, Length: 8
    - #4, Set: {'#6', '#8', '#5', '#1', '#7', '#3'}, Length: 6
    - #5, Set: {'#6', '#11', 'RAX', '#2', '#0', '#8', '#3', '#4', '#1', '#7'}, Length: 10
    - #6, Set: {'#11', 'RAX', '#2', '#0', '#8', '#5', '#3', '#4', '#1', '#7'}, Length: 10
    - #7, Set: {'#6', '#11', 'RAX', '#2', '#0', '#8', '#5', '#3', '#4', '#1'}, Length: 10
    - #8, Set: {'#6', '#11', '#2', '#0', '#5', '#3', '#4', '#1', '#7'}, Length: 9
    - #11, Set: {'#6', '#2', '#8', '#5', '#7', '#3'}, Length: 6
    - RAX, Set: {'#6', '#2', '#5', '#7'}, Length: 4

2. Simplify #11
    - #0, Set: {'#6', '#2', '#8', '#5', '#7'}, Length: 5
    - #1, Set: {'#6', '#2', '#8', '#5', '#3', '#4', '#7'}, Length: 7
    - #2, Set: {'#6', 'RAX', '#0', '#8', '#5', '#3', '#1', '#7'}, Length: 8
    - #3, Set: {'#6', '#2', '#8', '#5', '#4', '#1', '#7'}, Length: 7
    - #4, Set: {'#6', '#8', '#5', '#1', '#7', '#3'}, Length: 6
    - #5, Set: {'#6', 'RAX', '#2', '#0', '#8', '#3', '#4', '#1', '#7'}, Length: 9
    - #6, Set: {'RAX', '#2', '#0', '#8', '#5', '#3', '#4', '#1', '#7'}, Length: 9
    - #7, Set: {'#6', 'RAX', '#2', '#0', '#8', '#5', '#3', '#4', '#1'}, Length: 9
    - #8, Set: {'#6', '#2', '#0', '#5', '#3', '#4', '#1', '#7'}, Length: 8
    - RAX, Set: {'#6', '#2', '#5', '#7'}, Length: 4
3. Coalesce. #2和#4符合合并的要求
    - #0, Set: {'#6', '#2', '#8', '#5', '#7'}, Length: 5
    - #1, Set: {'#6', '#2', '#8', '#5', '#3', '#7'}, Length: 6
    - #2, Set: {'#6', 'RAX', '#0', '#8', '#5', '#3', '#1', '#7'}, Length: 8
    - #3, Set: {'#6', '#2', '#8', '#5', '#1', '#7'}, Length: 6
    - #5, Set: {'#6', 'RAX', '#2', '#0', '#8', '#3', '#1', '#7'}, Length: 8
    - #6, Set: {'RAX', '#2', '#0', '#8', '#5', '#3', '#1', '#7'}, Length: 8
    - #7, Set: {'#6', 'RAX', '#2', '#0', '#8', '#5', '#3', '#1'}, Length: 8
    - #8, Set: {'#6', '#2', '#0', '#5', '#3', '#1', '#7'}, Length: 7
    - RAX, Set: {'#6', '#2', '#5', '#7'}, Length: 4
4. Simplify #1, #3 符合要求
    - #0, Set: {'#6', '#2', '#8', '#5', '#7'}, Length: 5
    - #2, Set: {'#6', 'RAX', '#0', '#8', '#5', '#7'}, Length: 6
    - #5, Set: {'#6', 'RAX', '#2', '#0', '#8', '#7'}, Length: 6
    - #6, Set: {'RAX', '#2', '#0', '#8', '#5', '#7'}, Length: 6
    - #7, Set: {'#6', 'RAX', '#2', '#0', '#8', '#5'}, Length: 6
    - #8, Set: {'#6', '#2', '#0', '#5','#7'}, Length: 5
    - RAX, Set: {'#6', '#2', '#5', '#7'}, Length: 4
5. Simplify # 2, #5, #6, #7
6. Simplify #0
7. Coalesce #8, RAX

8. Simplify #8(RAX)

寄存器分配：

```
select:
RAX: #8(#RAX)
x1: #0,#1, #11,#9
x2: #7
x3: #6
x4: #5
x5:#2(#4), #12
x6: #3,#10
```