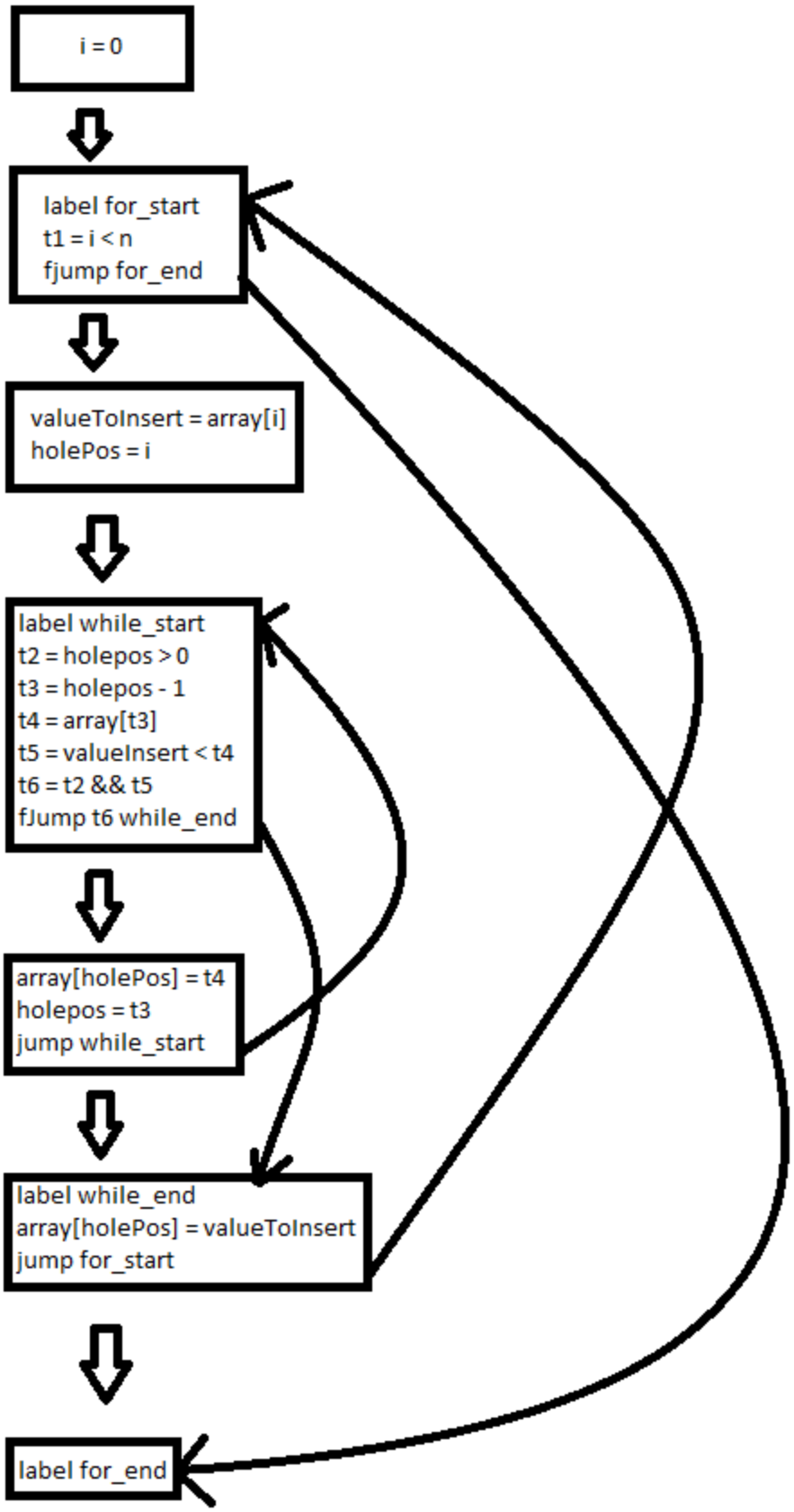
Part 1, Theory

Problem 1, Optimization

a)

```
i = 0
label for_start
t1 = i < n
fjump for_end
valueToInsert = array[i]
holePos = i
label while_start
t2 = holepos > 0
t3 = holepos - 1
t4 = array[t3]
t5 = valueInsert < t4
t6 = t2 && t5
fJump t6 while_end
array[holePos] = t4
holepos = t3
jump while_start
label while_end
array[holePos] = valueToInsert
i = i + 1
jump for_start
label for_end
```

b)



c)

Optimization 1:

Exchange holePos with i.
Optimization 2: Exchange holePos - 1 with t3 (i - 1)
Optimization 3: Exchange array[t3] with t4

```
i = 0
label for_start
t1 = i < n
fjump for_end
valueToInsert = array[i]
//holePos = i  <- removed this.
label while_start
t2 = i > 0
t3 = i - 1
t4 = array[t3]
t5 = valueInsert < t4
t6 = t2 && t5
fJump t6 while_end
array[i] = t4
//holepos = t3 <- removed this
jump while_start
label while_end
//array[t3] = valueToInsert <- removed this
t4 = valueToInsert
i = i + 1
jump for_start
label for_end
```