In [67]:
```
TAC = {"1": "count=0",
"2": "result=0",
"3": "if count > 20 GOTO 8",
"4": "count=count + 1",
"5": "increment = 2 * count",
"6": "result = result +increment",
"7": "GOTO 3",
"8": "end"}
```

In [68]:
```
TAC
```

Out[68]:
```
{'1': 'count=0',
 '2': 'result=0',
 '3': 'if count > 20 GOTO 8',
 '4': 'count=count + 1',
 '5': 'increment = 2 * count',
 '6': 'result = result +increment',
 '7': 'GOTO 3',
 '8': 'end'}
```

In [74]:
```
# 1ST, 3RD, 4TH, 8TH
LEADER_STMT = []
blockList = []
for k,v in TAC.items():
    if LEADER_STMT == []:
        LEADER_STMT.append((v,1))
        blockList.append(1);
    if v.__contains__('GOTO'):
        LEADER_STMT.append((TAC[v[-1]], int(v[-1])))
        blockList.append(int(v[-1]))
    if v.__contains__('if'):
#         print(int(k)+1)
        LEADER_STMT.append((TAC[str(int(k)+1)], int(k)+1))
        blockList.append(int(k) +1)
LEADER_STMT.sort(key = lambda x: x[1])
```

In [75]:
```
LEADER_STMT
```

Out[75]:
```
[('count=0', 1),
 ('if count > 20 GOTO 8', 3),
 ('count=count + 1', 4),
 ('end', 8)]
```

In [164]:
```
blockList = sorted(blockList)
blockList
```

Out[164]:
```
[1, 3, 4, 8]
```

In [192]:
```
blocks = {}
index = 1
for i in blockList:
    firstIndex = blockList.index(i)
    if firstIndex != len(blockList)-1:
        secondIndex = firstIndex+1
    else:
        secondIndex = firstIndex
    if firstIndex == blockList[-1] and firstIndex == secondIndex:
        blocks[f'B{index}'] = firstIndex
        index+=1
        break
    else:
        blocks[f'B{index}'] = (blockList[firstIndex], blockList[secondIndex]-1)
        index+=1
#     print(blockList[firstIndex], blockList[secondIndex]-1)
for k,v in blocks.items():
#     print(v)
    if v[0] == v[1]: # (3,3)
        blocks[k] = (v[0])
    if v[0] > v[1]: # (8,7)
        blocks[k] = (v[0])
```

In [193]:
```
blocks
```

Out[193]:
```
{'B1': (1, 2), 'B2': 3, 'B3': (4, 7), 'B4': 8}
```

In [194]:
```
LEADER_STMT
```
Out[194]:
```
[('count=0', 1),
 ('if count > 20 GOTO 8', 3),
 ('count=count + 1', 4),
 ('end', 8)]
```
In [195]:
```
TAC
```
Out[195]:
```
{'1': 'count=0',
 '2': 'result=0',
 '3': 'if count > 20 GOTO 8',
 '4': 'count=count + 1',
 '5': 'increment = 2 * count',
 '6': 'result = result +increment',
 '7': 'GOTO 3',
 '8': 'end'}
```
In [238]:
```
PFG = []
for k,v in TAC.items():
    if v.__contains__("if"):
        # 1 - > 2
        for key,val in blocks.items():
            if type(val) != int:
                if int(k)-1 in val or int(k) in val:
                    first = key
            if int(k) == val or int(k)-1 == val:
                second = key
        PFG.append((first, second))
        # 2 -> 3
        for key,val in blocks.items():
            if type(val) != int:
                if int(k)+1 in val or int(k) in val:
                    first = key
            if int(k) == val or int(k)+1 == val:
                second = key
        PFG.append((second, first))
    if v.__contains__("GOTO"):
        nextstmt = v.split("GOTO ")[-1]
d        for key,val in blocks.items():
            if type(val) != int:
                if int(k) in val or int(nextstmt) in val:
                    first = key
            if int(k) == val or int(nextstmt) == val:
                second = key
        print(first, second)
PFG
```
```
B3 B4
B3 B2
```
Out[238]:
```
[('B1', 'B2'), ('B2', 'B3')]
```
In [211]:
```
# B1 -> B2
# B2 -> B3
# B2 -> B4
# B3 -> B2
PFG = []

for k,v in TAC.items():
#     print(k,v)
    if v.startswith("if"):
        print(int(k)-1, int(k))
        nextBlock = int(k)+1
        print(int(k), nextBlock)
        print(blocks)
        for key,val in blocks.items():
            if type(val) != int:
                if int(k)-1 in val or int(k) in val:
                    first = key
            if int(k) == val or int(k)-1 == val:
                second = key
        PFG.append((first, second))
```
```
2 3
3 4
{'B1': (1, 2), 'B2': 3, 'B3': (4, 7), 'B4': 8}
```
In [210]:
```
PFG
```
Out[210]:
```
[('B1', 'B2')]
```

```
In [ ]:
```