cs3821 Assignment 2

Jennifer z5207930

April 2020

1 Question 4 (II)

1.1 Polynomial Register Algorithm

Given

```
N <- # Variables
E <- # Expressions
C <- Largest carnality (Number of Variables in an expression)
D <- Largest Interference (Number of Variables seen together at once)
Assumption: C <= D + 1
```

In the worst case, the D is 0 and we only have 1 register available. That means all N variables must be stored and loaded into a single register. In the best case, D is equal to N and hence every variable has a single register. For all the cases in between where we have more variables than registers available, we want to conserve the amount of loading occurring. Hence we want to prioritise certain variable according to their interference degree. Higher interference degree variables are stored in a single register while other registers can hold up to two to three other variables.
Let R be D + 1, the number of registers we have available.
Let S = R mod N, the amount of registers we want to only contain a single variable.
Let T = (R / N), the amount of variables each register will hold.

For example, consider the case where

```
a = b
c = d
e = f
```

Then we have 6 variables a, b, c, d, e, f but only interference 1. From earlier, R = 2, S = 0 and T = 3. Then we want to allocate each variable in the order in which the variables appear. In this case, a, c, e would be stored in register 1 and b, d, f would be stored in register 2.

In the case where S is not 0, we want to assign the first S variables with the highest interference in registers 1 to S. Then we would iterate through the remaining registers in the order in which they appear and store them into the remaining registers.

1.2 Worst Case Complexity

```
Assume: Variables are encoded as integers (0 ~ N-1)
Assume: G is an array which lets you test whether an array appears in the expression
```

The most complicated part would be to find the interference. Since we have G, and can test whether an array appears in O(1), we can find the interference deg with O(n). Then assign variables in O(n) by assigning the first S registers with the highest interference. Since S is always smaller than N, iterating through to find the variables with the highest interference would be a S * O(n) search, and therefore an overal O(n) search. Following that, it will still be O(n) as we allocate each variable into a register in the order they appear first.

Therefore overall, in the worst case, it will be O(n) complexity.

1.3 Proof

The algorithm always no more than D + 1 registers because we base it off the fact that D + 1 is the maximum number of registers available. We can calculate our storage since $N = (R - S) * T + S$ and therefore to store N variables we need no more than R registers where $R = D + 1$.