Code Analysis for Day

TCSS 342: Data Structures
HW2: String Mutation
Jesse Bannon

Methods called in Day

private ArrayList<Genome> quickSort(ArrayList<Genome> theList)

Description: Quicksorts the entire genome ArrayList. Let ***n*** be the number of genomes in the ArrayList. Average run time of this sorting method is $\log_2(\boldsymbol{n})$.

Value: $\sum_{i=0}^{\log(n)} n$

public Genome(Genome gene)

Description: Creates a new genome by copying an existing genome's properties into the new one. There are a total of three properties copied (String gene, double mutation rate, Random rand).

Value: c_{copy} (constant)

public void mutate()

Description: Tests a boolean based on mutation rate to add a character to the existing genome's string or delete a random character. Then converts the string into a character array to iterate through each character and randomly change it based on the same mutation rate boolean. Since a genome string is highly unlikely to go over the target string's length, we will consider this method constant.

Value: c_{mutate} (constant)

public void crossover(Genome other)

Description: Creates a new string by iterating through both genome strings at the same time. Each index has an equal chance of being the original genome string's character at that index or the other's. If one genome string is less than the other, there will be an equal chance to either append the char of the longer string at that index or break from the loop and assign the new string to the original genome's gene string. Since a genome string is highly unlikely to go over the target string's length, we will consider this method constant.

Value: $c_{crossover}$ (constant)

Runtime of Day

```
    void day()
    {
1       final int popSize = population.size();                                  2
2       population = quickSort(population);                                     log(n) Σ_{i=0} n
3       mostFit = population.get(0);                                            3
                                                                                LOOP
4       for (int i = popSize/2; i < popSize; i++)                                   1
        {
5           final Genome temp = new Genome(this.population.get(rand.nextInt(popSize/2)));   1 + c_copy
6           if (rand.nextBoolean()) {                                           2
7               temp.mutate();                                                  c_mutate
            } else {
8               temp.crossover(this.population.get(rand.nextInt(popSize/2)));   c_crossover + 2
9               temp.mutate();                                                  c_mutate
            }
10          population.set(i, temp);                                            1
        }                                                                       END
    }
```

Let **n** equal the amount of genomes within the ArrayList population.
Let **f**(**n**) represent the method day().

$$\mathbf{f(n)} = 2 + \sum_{i=0}^{\log(n)} n + 1 + \sum_{i=\lfloor n/2 \rfloor}^{n} \left(1+1+c_{copy}+2+\left(c_{mutate} \vee c_{crossover}+2+c_{mutate}\right)+1\right)$$

$$= 3 + n\log(n) + \frac{n}{2}\left(5 + c_{copy} + \left(c_{mutate} \text{ OR } c_{crossover} + 2 + c_{mutate}\right)\right)$$

$$= 3 + n\left(\log(n) + \frac{1}{2}\left(5 + c_{copy} + \left(c_{mutate} \text{ OR } c_{crossover} + 2 + c_{mutate}\right)\right)\right)$$

Let $\mathbf{c}_1 = \frac{1}{2}\left(5 + c_{copy} + \left(c_{mutate} \text{ OR } c_{crossover} + 2 + c_{mutate}\right)\right)$

$\mathbf{f(n)} = 3 + \mathbf{n}(\log(\mathbf{n}) + \mathbf{c}_1)$,

$\mathbf{f(n)} \in \mathbf{O}(\mathbf{n}\log(\mathbf{n}))$