**Iterations Lab Three**

**Document Id 4**

**In this section we worked through some introductory material for Working with Maps and inbuilt Java 8 Stream Algorithms**

**Questions**

**Code Fragment 1**

**Map<Integer, String> map = new HashMap<>();**

1 ) For Code Fragment 1 which expression fails to compile

1. Set<Integer> keys = map.keySet();
2. map.keySet().iterator().next();
3. new ArrayList(map.values());
4. List<String> values = map.values();
5. map.entrySet().iterator().next();

**Code Fragment 2**

**Map<String, List<Model>> byField1**

**= createModelList().stream()**

**.collect(Collectors.groupingBy(Model::getField1));**

2 ) For Code Fragment 2 which statement best describes the Algorithm

1. This acts as a reduction as it groups the values to a key
2. This acts as a shuffle as it groups the values to a key
3. This acts as an aggregation as it collects all the values by the key
4. This acts as a mapreduce as it groups the values to a key
5. None of the above

**Code Fragment 3**

**String joined = createModelList().stream()**

**.map(Model::getField1)**

**.collect(Collectors.joining(", "));**

3) For Code Fragment 3 which statement best describes the Algorithm

1. This acts as a reduction as it takes a collection to a constant
2. This acts as a mapping as it takes a value from a collection and transforms it
3. This acts as an aggregation as it collects all the values
4. This acts as a mapreduce as it groups the values to a key
5. None of the above

**Practical Matching Pattern Sequences with iterations**

Gene sequences are often mapped to binary sequences by combining the nucleotides

<https://arxiv.org/ftp/arxiv/papers/1208/1208.5713.pdf>

The alignment of the certain protein sequences can be measured in terms of a binary gap

<https://en.wikipedia.org/wiki/Sequence_alignment>

The above pattern matching of genetic data is just one example of the importance of pattern matching in sequence data

In this practical we look at an algorithm to match to a ‘gap’ pattern

Start with a binary number

**String binNumStr = Integer.toBinaryString(169);**  **// binNumStr ==== “10101001”**

We define the collection of binary gaps as being the sub sequences of successive zeros

**“10101001” -> [ “0”, “0”, “00”]**

For the above collection  **[ “0”, “0”, “00”]** the max binary gap is of size 2, **“00”**

Create an algorithm that finds the maximum binary gap for a given binary number