Michael Thomas

```
17      default IDnaStrand cutAndSplice(String enzyme, String splicee) {
18          String search = this.toString();
19          IDnaStrand ret = getInstance(source: "");
20          // Splits dna strand by enzyme, leaving empty strings
21          // in case of leading, repeating, or trailing enzymes
22          String[] fragments = search.split(enzyme+"{1}", -1);
23          for (int i=0; i<fragments.length-1; i++) {    // splicing in
24              ret.append(fragments[i]);
25              ret.append(splicee);
26          }
27          ret.append(fragments[fragments.length-1]);    // adding last fragment
28          return ret;
29      }
```

```
61      @Override
62      public IDnaStrand append(String dna) {
63          myInfo = myInfo + dna;
64          myAppends++;
65          return this;
66      }
```

Question 1: What is the big O asymptotic runtime complexity of cutAndSplice when using StringStrand, in terms of N, b, and S? Justify your answer in theory, referencing the implementation, and empirically, by reporting your results from running DNABenchmark.

The big O asymptotic runtime complexity of toString() is O(1) and it runs once. The big O asymptotic runtime complexity of getInstance() is O(1), and it runs once. The big O asymptotic runtime complexity of split() on line 22 is O(N), and it runs once. This method will lead fragments to be of length b+1, so the forloop on line 23 will run b times. The Length of fragments[i] on line 24 will be N/b, because assuming the length of the enzymes is negligible there are N characters divided by b+1 fragments, which is close to N/b. The big O asymptotic runtime complexity of the append() method on line 24 and the append() method on line 25 is an arithetic sequence with b terms and each term goes from 0 + 2N/b + S to N + bS. Therefore the overall big O asymptotic runtime complexity of cutAndSplice is O(bN + (b^2)S)), with all smaller terms being negligible.

```
dna length = 320,160
cutting at enzyme gaattc
-------------------------------------------------------------------------
Class             dna,N   splicee,S        recomb  time(ms)  breaks,b
-------------------------------------------------------------------------
StringStra:     320,160      10,000       769,890        30        45
StringStra:     320,160      20,000     1,219,890        27        45
StringStra:     320,160      40,000     2,119,890        24        45
StringStra:     320,160      80,000     3,919,890        39        45
StringStra:     320,160     160,000     7,519,890        77        45
StringStra:     320,160     320,000    14,719,890       157        45
StringStra:     320,160     640,000    29,119,890       358        45
StringStra:     320,160   1,280,000    57,919,890       793        45
StringStra:     320,160      10,000       769,890         8        45
StringStra:     640,320      10,000     1,539,780        35        90
StringStra:   1,280,640      10,000     3,079,560       123       180
StringStra:   2,561,280      10,000     6,159,120       499       360
StringStra:   5,122,560      10,000    12,318,240     2,194       720
```

Empirically, as S doubles, the runtime approximately doubles, showing a linear relationship with S. As b multiplies by 16 and N multiplies by 16, runtime multiplies by approximately 274, which is close to 16^2(b^2) + 16(N).

Question 2: What is the big O asymptotic runtime complexity of cutAndSplice when using StringBuilderStrand, in terms of N, b, and S? Justify your answer in theory, referencing the implementation, and empirically, by reporting your results from running DNABenchmark.

O(N + bS)

The big O asymptotic runtime complexity of toString() and getInstance() is O(1). As before, the big O asymptotic runtime complexity of split() is O(N). Then fragments.length is b + 1 and the forloop on line 23 runs b times. The big O asymptotic runtime complexity of append() on line 25 is O(S) and it runs b times. The big O asymptotic runtime complexity of append() on line 25 is O(N/b) and it runs b times. This

results in an overall big O asymptotic runtime complexity of O(bS + N).

```
dna length = 320,160
cutting at enzyme gaattc
-------------------------------------------------------------------------
Class             dna,N   splicee,S        recomb  time(ms)  breaks,b
-------------------------------------------------------------------------
StringBuil:     320,160      10,000       769,890         5        45
StringBuil:     320,160      20,000     1,219,890         2        45
StringBuil:     320,160      40,000     2,119,890         3        45
StringBuil:     320,160      80,000     3,919,890         3        45
StringBuil:     320,160     160,000     7,519,890         4        45
StringBuil:     320,160     320,000    14,719,890         8        45
StringBuil:     320,160     640,000    29,119,890        13        45
StringBuil:     320,160   1,280,000    57,919,890        15        45
StringBuil:     320,160      10,000       769,890         1        45
StringBuil:     640,320      10,000     1,539,780         3        90
StringBuil:   1,280,640      10,000     3,079,560         5       180
StringBuil:   2,561,280      10,000     6,159,120        10       360
StringBuil:   5,122,560      10,000    12,318,240        20       720
```

Empirically, we see that as S doubles, time roughly doubles, showing linear time complexity. We also see that as b and N double, the time doubles, which corresponds with O(bS + N) where b and N have linear time complexity.

Question 3: If each character of a String takes 1 byte of memory to store, about how much total memory is necessary to store the result of a cutAndSplice operation on a StringStrand object? Express your answer in terms of N, b, and S. Would the result take more or less memory if using a StringBuilderStrand object? Briefly explain.

The required memory for the result would be N + s - bE bytes where E is the length of the enzyme. It would take more memory using a StringBuilderStrand, because the StringStrand object points to the same spot in memory for each splice, whereas the StringBuilderObject allocates memory for each splice.

Question 4: What is the big O asymptotic runtime complexity of cutAndSplice when using LinkStrand in terms of N, b, and S? Justify your answer in theory, referencing the implementation, and empirically, by reporting your results from running DNABenchmark.

O(N + b)

The big O asymptotic runtime complexity of getInstance(), and append() is O(1). The big O asymptotic runtime complexity of toString() is O(N). As before, the big O asymptotic runtime complexity of split() is O(N). Then fragments.length is b + 1 and the forloop on line 23 runs b times. This results in an overall big O asymptotic runtime complexity of O(b + N).

```
dna length = 320,160
cutting at enzyme gaattc
--------------------------------------------------------------------------
Class             dna,N   splicee,S        recomb  time(ms)  breaks,b
--------------------------------------------------------------------------
LinkStrand:     320,160      10,000       769,890         3        45
LinkStrand:     320,160      20,000     1,219,890         3        45
LinkStrand:     320,160      40,000     2,119,890         1        45
LinkStrand:     320,160      80,000     3,919,890         1        45
LinkStrand:     320,160     160,000     7,519,890         1        45
LinkStrand:     320,160     320,000    14,719,890         1        45
LinkStrand:     320,160     640,000    29,119,890         1        45
LinkStrand:     320,160   1,280,000    57,919,890         1        45
LinkStrand:     320,160      10,000       769,890         1        45
LinkStrand:     640,320      10,000     1,539,780         2        90
LinkStrand:   1,280,640      10,000     3,079,560         6       180
LinkStrand:   2,561,280      10,000     6,159,120         9       360
LinkStrand:   5,122,560      10,000    12,318,240        18       720
```

Empirically, we see that S has no effect on runtime. We also see that as b and N double, the time doubles, which corresponds with O(b + N) where b and N have linear time complexity.

Question 5: If each character of a String takes 1 byte of memory to store, and each reference to a node takes 8 bytes of memory to store, about about how much total memory is necessary to store the result of a cutAndSplice operation on a LinkStrand object? Express your answer in terms of N, b, and S. Briefly explain your answer, refereincing the implementation of LinkStrand.

On line 19 getInstance is added to ret, which is 8 bytes. The forloop runs b times and each time it creates two nodes, 16 bytes, and adds strings length N/b and s. Then the last fragment is added for a total memory of b(16 + N/b + s) + N/b bytes.