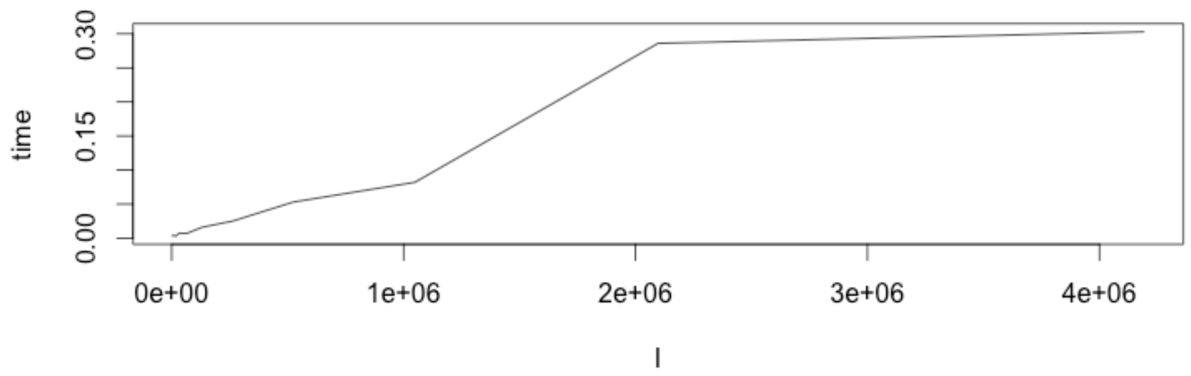
DNA Analysis

Part 1:

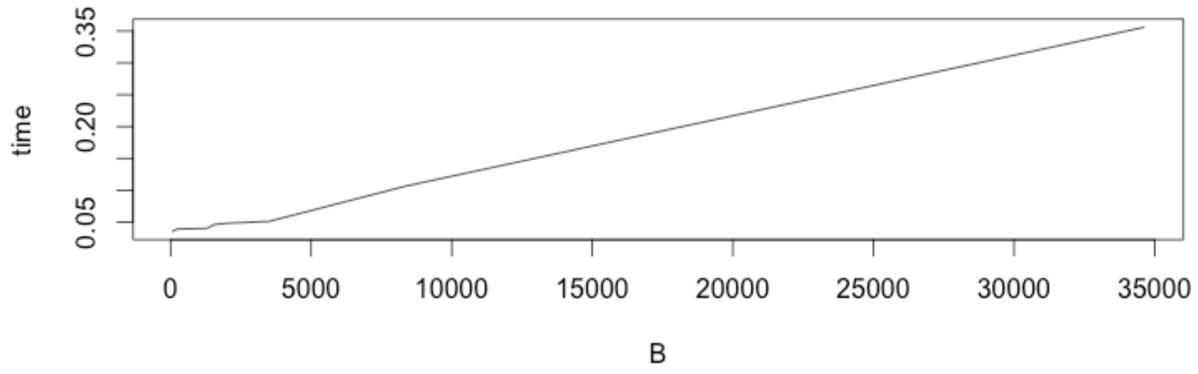
Fixed n, B and vary I.
n=320160, B=90



```
Call:
lm(formula = time ~ I)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.049384 -0.006646 -0.004974 -0.003990  0.105657 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 8.302e-03  1.128e-02   0.736    0.477    
I           8.204e-08  8.398e-09   9.768 9.34e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03567 on 11 degrees of freedom
Multiple R-squared:  0.8966,  Adjusted R-squared:  0.8872 
F-statistic: 95.42 on 1 and 11 DF,  p-value: 9.339e-07
```

The test shows that runtime and I are linearly related, when n and B are fixed.

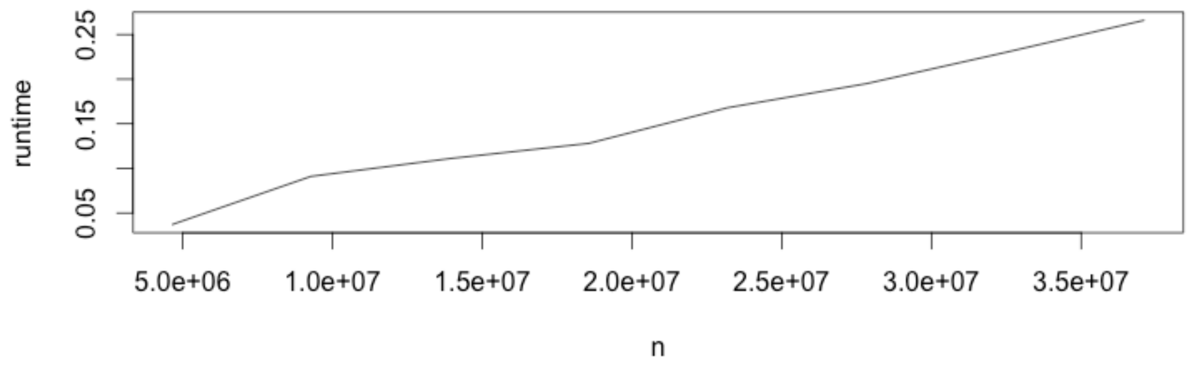
Fixed n, I and vary B.
n=4639221, I=8192
I change B by changing the String enzyme in Benchmark.



```
Call:
lm(formula = time ~ B)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.0115322 -0.0017128  0.0006737  0.0025342  0.0070192 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.967e-02  2.471e-03   12.01 2.02e-05 ***
B           9.377e-06  1.946e-07   48.19 5.36e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.00602 on 6 degrees of freedom
Multiple R-squared:  0.9974,  Adjusted R-squared:  0.997 
F-statistic:  2322 on 1 and 6 DF,  p-value: 5.356e-09
```

The test shows that runtime and B are linearly related, when n and I are fixed.

Fixed B,I and vary n.
B=1290, I=8192.
I change n by making copies of the original DNA with nucleotide t replaced by nucleotide a, and add the new copy to the original DNA.

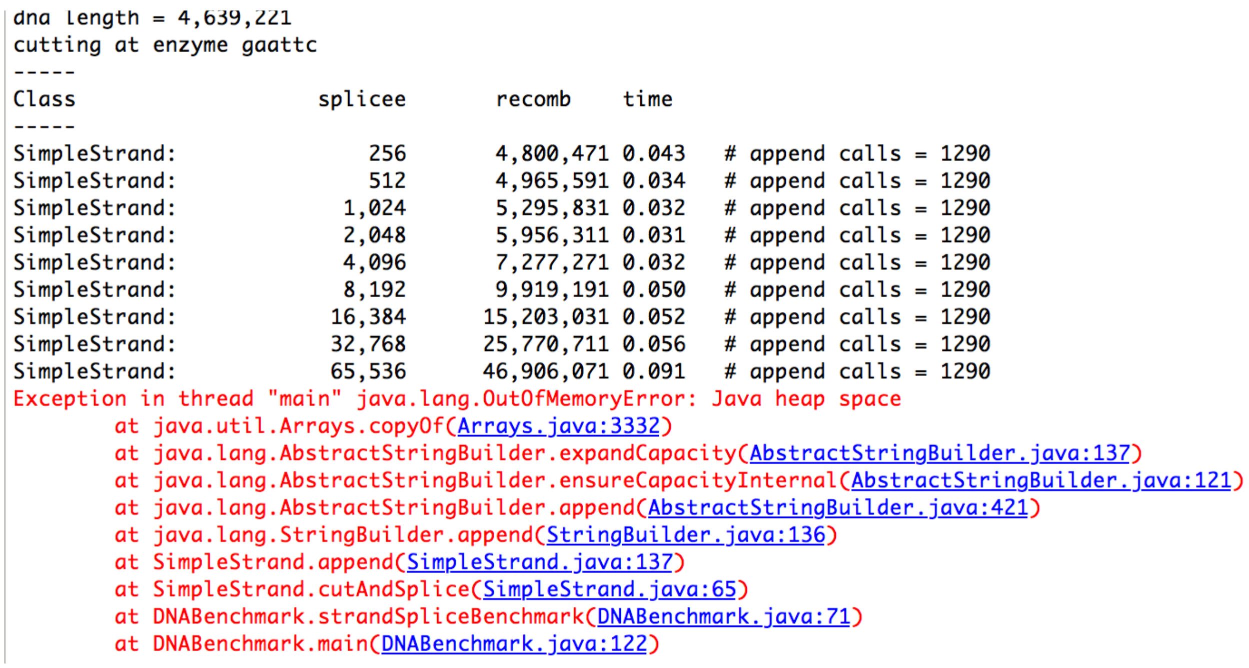


```
Call:
lm(formula = t ~ n)

Residuals:
     Min       1Q    Median       3Q       Max
-0.009833 -0.005458 -0.000500  0.004208  0.014833

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.450e-02  6.720e-03   2.158   0.0743 .
n           6.646e-09  2.869e-10  23.168 4.24e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.008625 on 6 degrees of freedom
Multiple R-squared:  0.9889,  Adjusted R-squared:  0.9871
F-statistic: 536.8 on 1 and 6 DF,  p-value: 4.239e-07
```

The test shows that runtime and n are linearly related, when B and I are fixed.

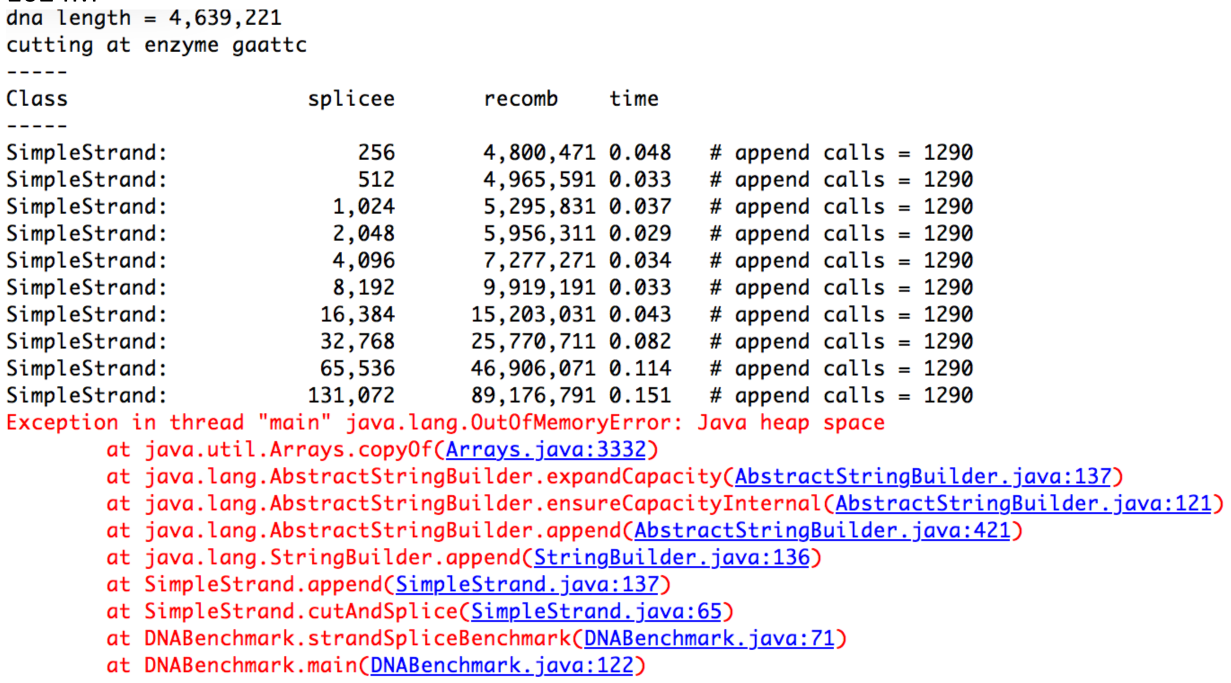
Part 2:

512M



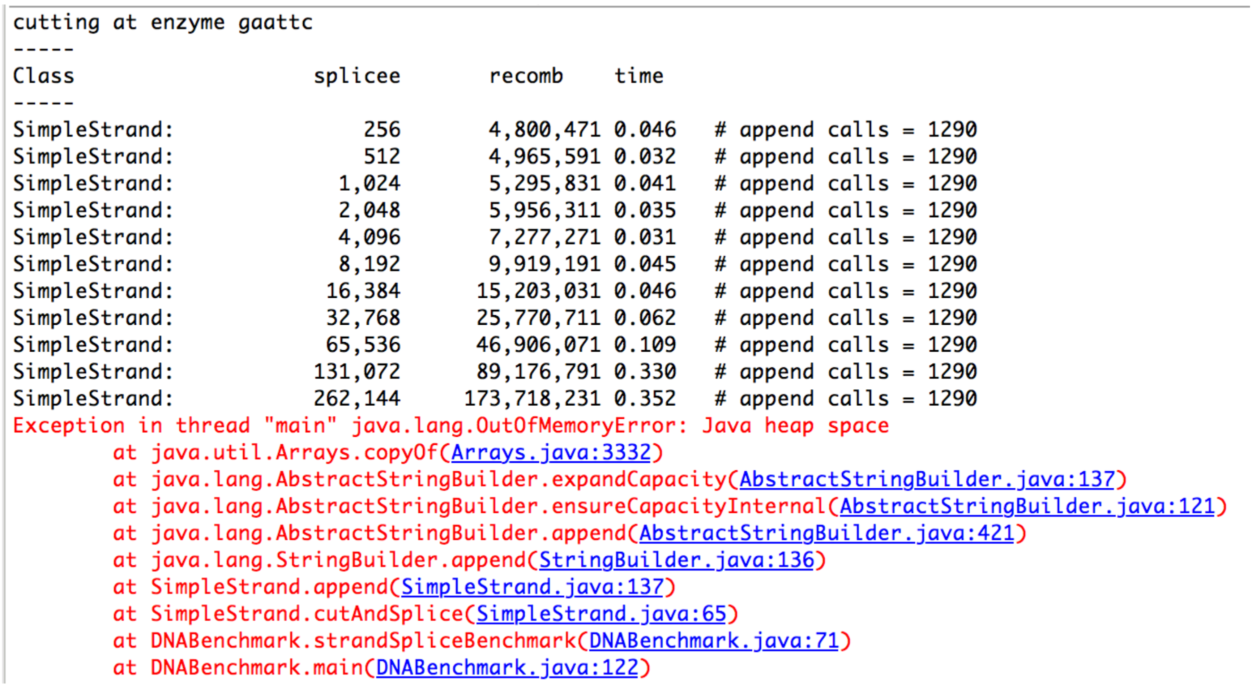
```
dna length = 4,639,221
cutting at enzyme gaattc
-----
Class                   splicee       recomb    time
-----
SimpleStrand:               256       4,800,471 0.043   # append calls = 1290
SimpleStrand:               512       4,965,591 0.034   # append calls = 1290
SimpleStrand:             1,024       5,295,831 0.032   # append calls = 1290
SimpleStrand:             2,048       5,956,311 0.031   # append calls = 1290
SimpleStrand:             4,096       7,277,271 0.032   # append calls = 1290
SimpleStrand:             8,192       9,919,191 0.050   # append calls = 1290
SimpleStrand:            16,384      15,203,031 0.052   # append calls = 1290
SimpleStrand:            32,768      25,770,711 0.056   # append calls = 1290
SimpleStrand:            65,536      46,906,071 0.091   # append calls = 1290
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
	at java.util.Arrays.copyOf(Arrays.java:3332)
	at java.lang.AbstractStringBuilder.expandCapacity(AbstractStringBuilder.java:137)
	at java.lang.AbstractStringBuilder.ensureCapacityInternal(AbstractStringBuilder.java:121)
	at java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:421)
	at java.lang.StringBuilder.append(StringBuilder.java:136)
	at SimpleStrand.append(SimpleStrand.java:137)
	at SimpleStrand.cutAndSplice(SimpleStrand.java:65)
	at DNABenchmark.strandSpliceBenchmark(DNABenchmark.java:71)
	at DNABenchmark.main(DNABenchmark.java:122)
```

1024M



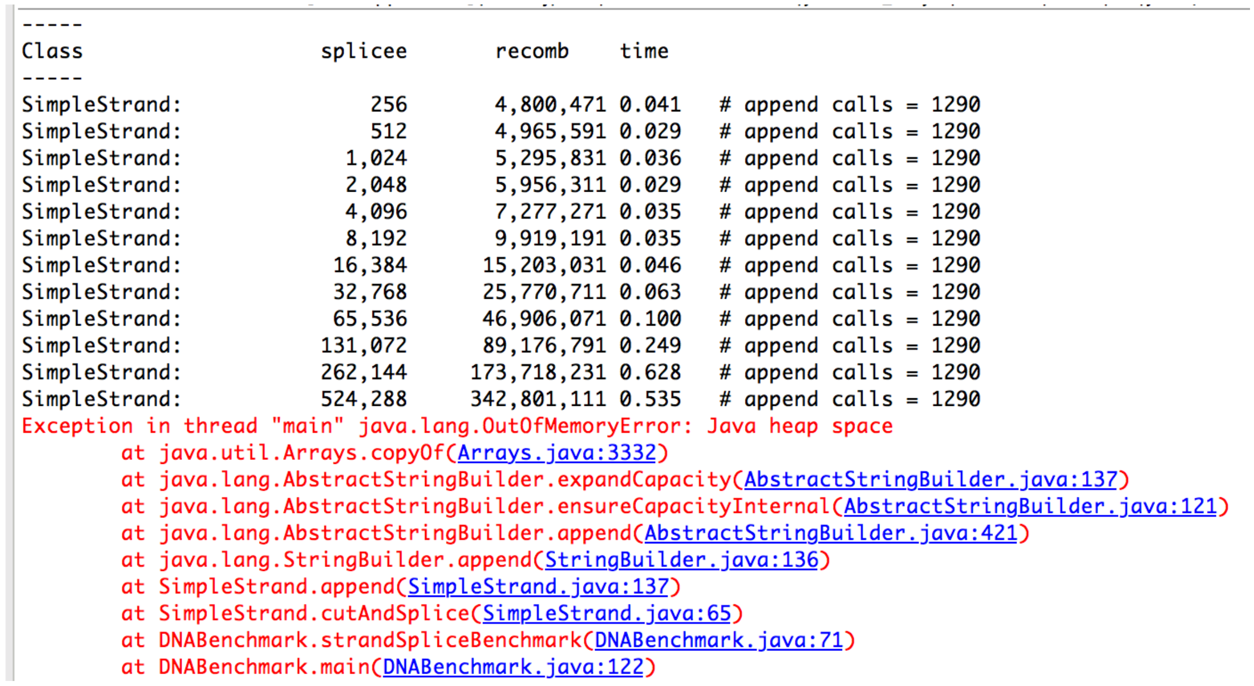
```
dna length = 4,639,221
cutting at enzyme gaattc
-----
Class                   splicee       recomb    time
-----
SimpleStrand:               256       4,800,471 0.048   # append calls = 1290
SimpleStrand:               512       4,965,591 0.033   # append calls = 1290
SimpleStrand:             1,024       5,295,831 0.037   # append calls = 1290
SimpleStrand:             2,048       5,956,311 0.029   # append calls = 1290
SimpleStrand:             4,096       7,277,271 0.034   # append calls = 1290
SimpleStrand:             8,192       9,919,191 0.033   # append calls = 1290
SimpleStrand:            16,384      15,203,031 0.043   # append calls = 1290
SimpleStrand:            32,768      25,770,711 0.082   # append calls = 1290
SimpleStrand:            65,536      46,906,071 0.114   # append calls = 1290
SimpleStrand:           131,072      89,176,791 0.151   # append calls = 1290
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
	at java.util.Arrays.copyOf(Arrays.java:3332)
	at java.lang.AbstractStringBuilder.expandCapacity(AbstractStringBuilder.java:137)
	at java.lang.AbstractStringBuilder.ensureCapacityInternal(AbstractStringBuilder.java:121)
	at java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:421)
	at java.lang.StringBuilder.append(StringBuilder.java:136)
	at SimpleStrand.append(SimpleStrand.java:137)
	at SimpleStrand.cutAndSplice(SimpleStrand.java:65)
	at DNABenchmark.strandSpliceBenchmark(DNABenchmark.java:71)
	at DNABenchmark.main(DNABenchmark.java:122)
```

2048M



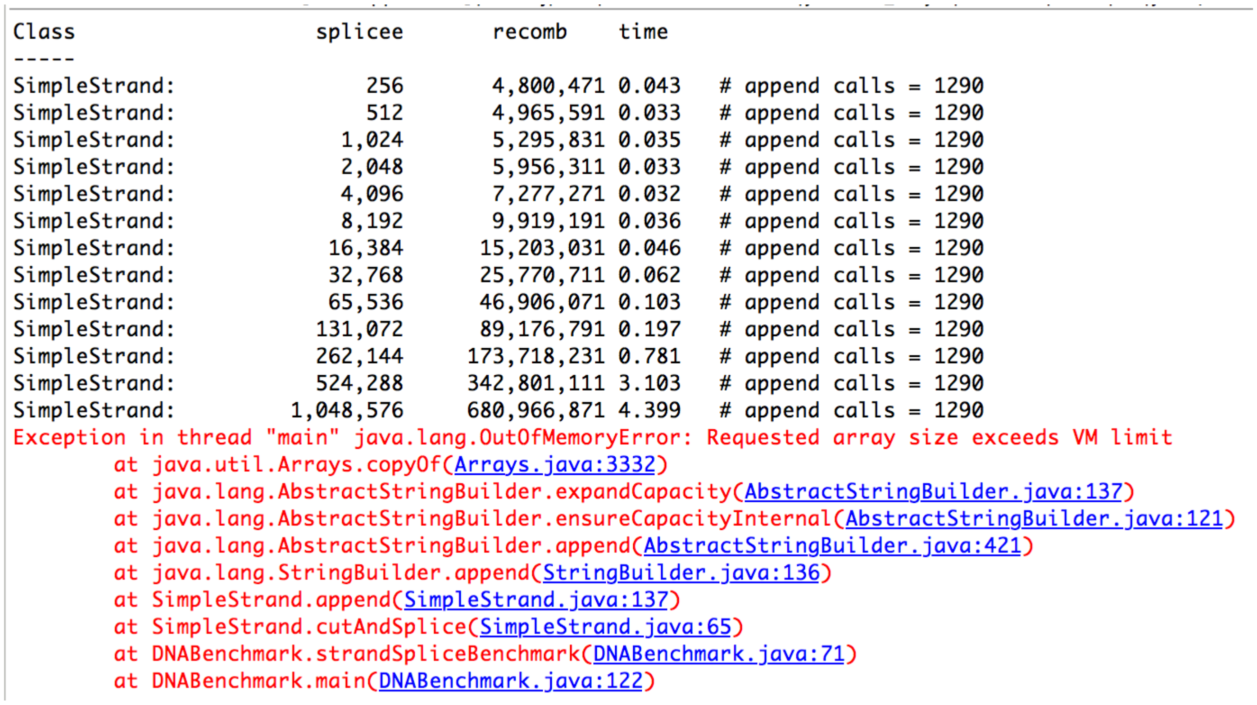
```
cutting at enzyme gaattc
-----
Class                   splicee       recomb    time
-----
SimpleStrand:               256      4,800,471 0.046   # append calls = 1290
SimpleStrand:               512      4,965,591 0.032   # append calls = 1290
SimpleStrand:             1,024      5,295,831 0.041   # append calls = 1290
SimpleStrand:             2,048      5,956,311 0.035   # append calls = 1290
SimpleStrand:             4,096      7,277,271 0.031   # append calls = 1290
SimpleStrand:             8,192      9,919,191 0.045   # append calls = 1290
SimpleStrand:            16,384     15,203,031 0.046   # append calls = 1290
SimpleStrand:            32,768     25,770,711 0.062   # append calls = 1290
SimpleStrand:            65,536     46,906,071 0.109   # append calls = 1290
SimpleStrand:           131,072     89,176,791 0.330   # append calls = 1290
SimpleStrand:           262,144    173,718,231 0.352   # append calls = 1290
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
	at java.util.Arrays.copyOf(Arrays.java:3332)
	at java.lang.AbstractStringBuilder.expandCapacity(AbstractStringBuilder.java:137)
	at java.lang.AbstractStringBuilder.ensureCapacityInternal(AbstractStringBuilder.java:121)
	at java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:421)
	at java.lang.StringBuilder.append(StringBuilder.java:136)
	at SimpleStrand.append(SimpleStrand.java:137)
	at SimpleStrand.cutAndSplice(SimpleStrand.java:65)
	at DNABenchmark.strandSpliceBenchmark(DNABenchmark.java:71)
	at DNABenchmark.main(DNABenchmark.java:122)
```

4096M



```
-----
Class                   splicee       recomb    time
-----
SimpleStrand:               256      4,800,471 0.041   # append calls = 1290
SimpleStrand:               512      4,965,591 0.029   # append calls = 1290
SimpleStrand:             1,024      5,295,831 0.036   # append calls = 1290
SimpleStrand:             2,048      5,956,311 0.029   # append calls = 1290
SimpleStrand:             4,096      7,277,271 0.035   # append calls = 1290
SimpleStrand:             8,192      9,919,191 0.035   # append calls = 1290
SimpleStrand:            16,384     15,203,031 0.046   # append calls = 1290
SimpleStrand:            32,768     25,770,711 0.063   # append calls = 1290
SimpleStrand:            65,536     46,906,071 0.100   # append calls = 1290
SimpleStrand:           131,072     89,176,791 0.249   # append calls = 1290
SimpleStrand:           262,144    173,718,231 0.628   # append calls = 1290
SimpleStrand:           524,288    342,801,111 0.535   # append calls = 1290
Exception in thread "main" java.lang.OutOfMemoryError: Java heap space
	at java.util.Arrays.copyOf(Arrays.java:3332)
	at java.lang.AbstractStringBuilder.expandCapacity(AbstractStringBuilder.java:137)
	at java.lang.AbstractStringBuilder.ensureCapacityInternal(AbstractStringBuilder.java:121)
	at java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:421)
	at java.lang.StringBuilder.append(StringBuilder.java:136)
	at SimpleStrand.append(SimpleStrand.java:137)
	at SimpleStrand.cutAndSplice(SimpleStrand.java:65)
	at DNABenchmark.strandSpliceBenchmark(DNABenchmark.java:71)
	at DNABenchmark.main(DNABenchmark.java:122)
```

8192M


```
Class                       splicee       recomb    time
-----
SimpleStrand:                   256       4,800,471 0.043   # append calls = 1290
SimpleStrand:                   512       4,965,591 0.033   # append calls = 1290
SimpleStrand:                 1,024       5,295,831 0.035   # append calls = 1290
SimpleStrand:                 2,048       5,956,311 0.033   # append calls = 1290
SimpleStrand:                 4,096       7,277,271 0.032   # append calls = 1290
SimpleStrand:                 8,192       9,919,191 0.036   # append calls = 1290
SimpleStrand:                16,384      15,203,031 0.046   # append calls = 1290
SimpleStrand:                32,768      25,770,711 0.062   # append calls = 1290
SimpleStrand:                65,536      46,906,071 0.103   # append calls = 1290
SimpleStrand:               131,072      89,176,791 0.197   # append calls = 1290
SimpleStrand:               262,144     173,718,231 0.781   # append calls = 1290
SimpleStrand:               524,288     342,801,111 3.103   # append calls = 1290
SimpleStrand:             1,048,576     680,966,871 4.399   # append calls = 1290
Exception in thread "main" java.lang.OutOfMemoryError: Requested array size exceeds VM limit
        at java.util.Arrays.copyOf(Arrays.java:3332)
        at java.lang.AbstractStringBuilder.expandCapacity(AbstractStringBuilder.java:137)
        at java.lang.AbstractStringBuilder.ensureCapacityInternal(AbstractStringBuilder.java:121)
        at java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:421)
        at java.lang.StringBuilder.append(StringBuilder.java:136)
        at SimpleStrand.append(SimpleStrand.java:137)
        at SimpleStrand.cutAndSplice(SimpleStrand.java:65)
        at DNABenchmark.strandSpliceBenchmark(DNABenchmark.java:71)
        at DNABenchmark.main(DNABenchmark.java:122)
```

I determined the power-of-two string I can use in both memory sizes by checking the last line above the “Exception in thread “main” java.lang.OutOfMemoryError: Java heap space” message.

For 512M of heap-size, the largest power-of-two string is of length 65,536. And the time is 0.091.

For 1024M of heap-size, it can fit in the next power-of-two string. The largest power-of-two string is of length 131,072. And the time is 0.151.

For 2048M of heap-size, it can fit in the next power-of-two string. The largest power-of-two string is of length 262,144. And the time is 0.352.

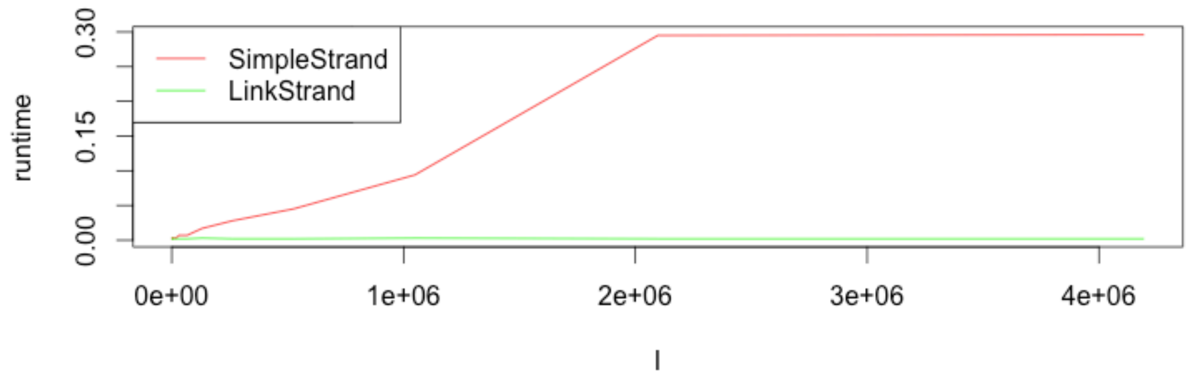
For 4096M of heap-size, it can fit in the next power-of-two string. The largest power-of-two string is of length 524,288. And the time is 0.535.

For 8192M of heap-size, it can fit in the next power-of-two string. The largest power-of-two string is of length 1,048,576. And the time is 4.399.

For 16384M of heap-size, there is no improvement. The largest power-of-two string is of length 1,048,576. And the time is 11.629.

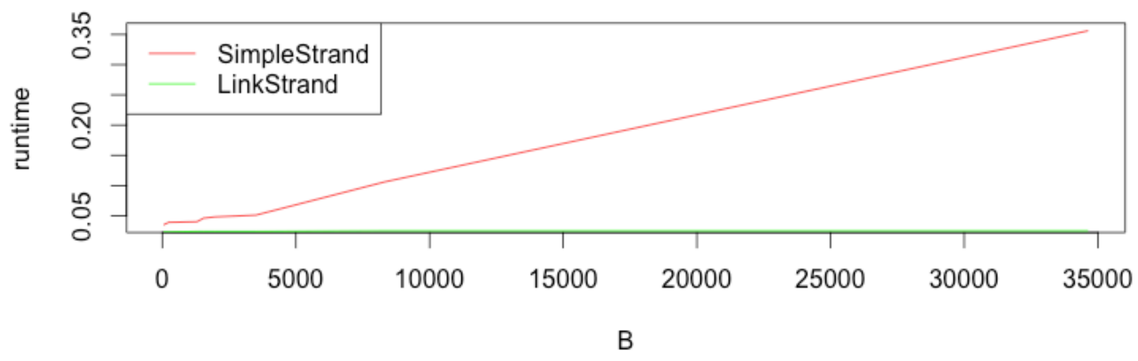
Part 3:

Fixed n and B, vary I



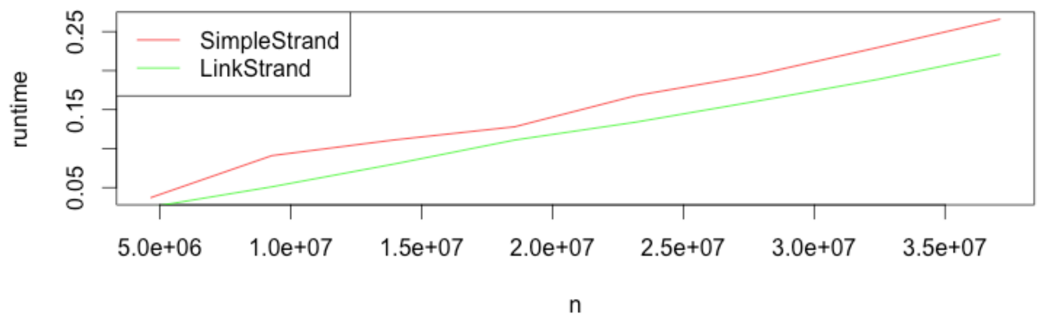
When I varies, the LinkStrand is more efficient. And the runtime is in O(1).

Fixed n and I, vary B



When B varies, the LinkStrand is more efficient. And the runtime is in O(1).

Fixed B and I, vary n



When n varies, the LinkStrand is slightly more efficient. And the runtime is in O(n).