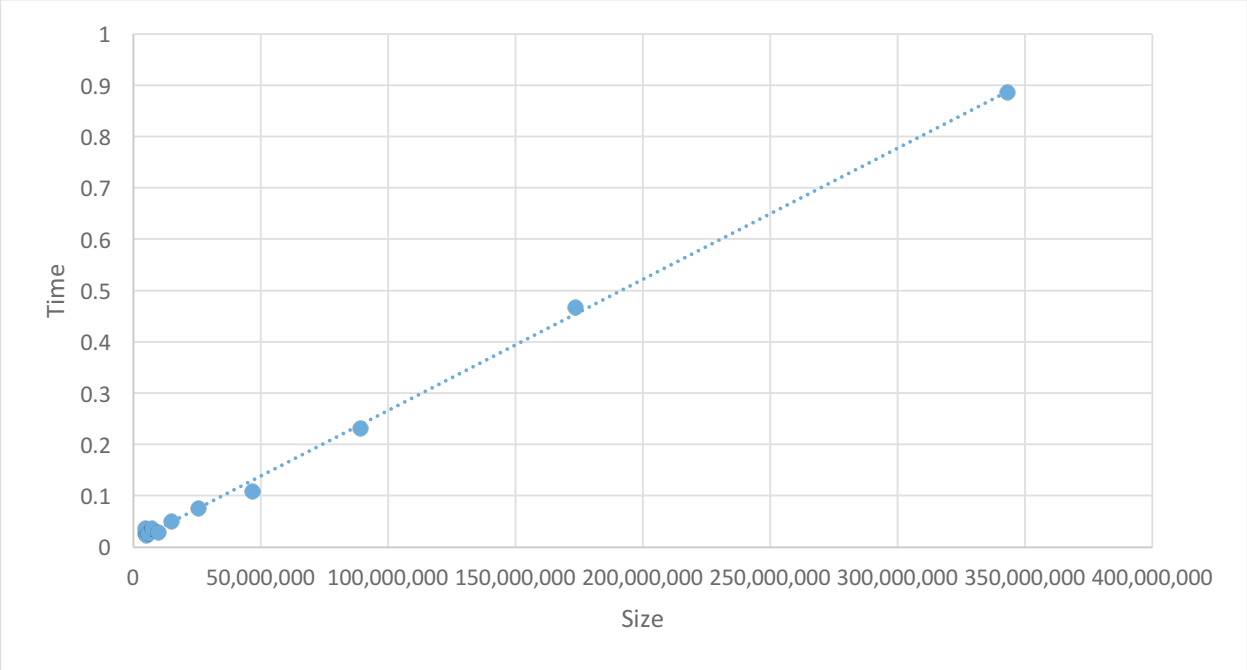
Tianyi Mu
CS201
DNA

Analysis Part 1: SimpleStrand

dna length = 4,639,221
cutting at enzyme gaattc

Class	splicee	recombtime		
SimpleStrand:	256	4,800,471	0.025	# append calls = 1290
SimpleStrand:	512	4,965,591	0.037	# append calls = 1290
SimpleStrand:	1,024	5,295,831	0.023	# append calls = 1290
SimpleStrand:	2,048	5,956,311	0.026	# append calls = 1290
SimpleStrand:	4,096	7,277,271	0.036	# append calls = 1290
SimpleStrand:	8,192	9,919,191	0.028	# append calls = 1290
SimpleStrand:	16,384	15,203,031	0.049	# append calls = 1290
SimpleStrand:	32,768	25,770,711	0.074	# append calls = 1290
SimpleStrand:	65,536	46,906,071	0.108	# append calls = 1290
SimpleStrand:	131,072	89,176,791	0.232	# append calls = 1290
SimpleStrand:	262,144	173,718,231	0.467	# append calls = 1290
SimpleStrand:	524,288	342,801,111	0.885	# append calls = 1290



Based on the output of DNABenchmark, it is clear that the time increases as the size of recombined strand increases.

This is the case because appending to a StringBuilder takes O(n). When appending new strings with StringBuilder, the StringBuilder append character by character. Thus when returning the recombined object, the StringBuilder went through every character. So the running time is O(N).

Analysis Part 2: SimpleStrand

Heap size	Maximum splice size
512	65,536
1024	131,072
2048	262,144
4096	524,288
8192	1,048,576
16384	1,048,576

The maximum spice size is 1,048,576.

Analysis Part 3: LinkStrand

For ecoli.txt

dna length = 4,639,221
cutting at enzyme gaattc

Class	splicee	recombtime		
LinkStrand:	256	4,800,471	0.021	# append calls = 1290
LinkStrand:	512	4,965,591	0.022	# append calls = 1290
LinkStrand:	1,024	5,295,831	0.022	# append calls = 1290
LinkStrand:	2,048	5,956,311	0.020	# append calls = 1290
LinkStrand:	4,096	7,277,271	0.021	# append calls = 1290
LinkStrand:	8,192	9,919,191	0.022	# append calls = 1290
LinkStrand:	16,384	15,203,031	0.023	# append calls = 1290
LinkStrand:	32,768	25,770,711	0.024	# append calls = 1290
LinkStrand:	65,536	46,906,071	0.025	# append calls = 1290
LinkStrand:	131,072	89,176,791	0.024	# append calls = 1290
LinkStrand:	262,144	173,718,231	0.026	# append calls = 1290
LinkStrand:	524,288	342,801,111	0.021	# append calls = 1290
LinkStrand:	1,048,576	680,966,871	0.021	# append calls = 1290
LinkStrand:	2,097,152	1,357,298,391	0.023	# append calls = 1290
LinkStrand:	4,194,304	2,709,961,431	0.021	# append calls = 1290
LinkStrand:	8,388,608	5,415,287,511	0.020	# append calls = 1290
LinkStrand:	16,777,216	10,825,939,671	0.023	# append calls = 1290
LinkStrand:	33,554,432	21,647,243,991	0.019	# append calls = 1290
LinkStrand:	67,108,864	43,289,852,631	0.021	# append calls = 1290
LinkStrand:	134,217,728	86,575,069,911	0.020	# append calls = 1290
LinkStrand:	268,435,456	173,145,504,471	0.020	# append calls = 1290
LinkStrand:	536,870,912	346,286,373,591	0.024	# append calls = 1290

Average time is: 0.021954545

For ecoli_2x.txt

dna length = 9,278,442
cutting at enzyme gaattc

Class	splicee	recombtime		
LinkStrand:	256	9,600,942	0.042	# append calls = 2580
LinkStrand:	512	9,931,182	0.045	# append calls = 2580
LinkStrand:	1,024	10,591,662	0.041	# append calls = 2580

LinkStrand:	2,048	11,912,622	0.041	# append calls = 2580
LinkStrand:	4,096	14,554,542	0.041	# append calls = 2580
LinkStrand:	8,192	19,838,382	0.043	# append calls = 2580
LinkStrand:	16,384	30,406,062	0.045	# append calls = 2580
LinkStrand:	32,768	51,541,422	0.047	# append calls = 2580
LinkStrand:	65,536	93,812,142	0.046	# append calls = 2580
LinkStrand:	131,072	178,353,582	0.047	# append calls = 2580
LinkStrand:	262,144	347,436,462	0.046	# append calls = 2580
LinkStrand:	524,288	685,602,222	0.047	# append calls = 2580
LinkStrand:	1,048,576	1,361,933,742	0.062	# append calls = 2580
LinkStrand:	2,097,152	2,714,596,782	0.041	# append calls = 2580
LinkStrand:	4,194,304	5,419,922,862	0.041	# append calls = 2580
LinkStrand:	8,388,608	10,830,575,022	0.046	# append calls = 2580
LinkStrand:	16,777,216	21,651,879,342	0.040	# append calls = 2580
LinkStrand:	33,554,432	43,294,487,982	0.048	# append calls = 2580
LinkStrand:	67,108,864	86,579,705,262	0.041	# append calls = 2580
LinkStrand:	134,217,728	173,150,139,822	0.047	# append calls = 2580
LinkStrand:	268,435,456	346,291,008,942	0.041	# append calls = 2580
LinkStrand:	536,870,912	692,572,747,182	0.041	# append calls = 2580

Average time is: 0.0445

For ecoli_3x.txt

dna length = 13,917,663
cutting at enzyme gaattc

Class	splicee	recombtime		
LinkStrand:	256	14,401,413	0.064	# append calls = 3870
LinkStrand:	512	14,896,773	0.062	# append calls = 3870
LinkStrand:	1,024	15,887,493	0.061	# append calls = 3870
LinkStrand:	2,048	17,868,933	0.061	# append calls = 3870
LinkStrand:	4,096	21,831,813	0.062	# append calls = 3870
LinkStrand:	8,192	29,757,573	0.061	# append calls = 3870
LinkStrand:	16,384	45,609,093	0.061	# append calls = 3870
LinkStrand:	32,768	77,312,133	0.068	# append calls = 3870
LinkStrand:	65,536	140,718,213	0.083	# append calls = 3870
LinkStrand:	131,072	267,530,373	0.061	# append calls = 3870
LinkStrand:	262,144	521,154,693	0.060	# append calls = 3870
LinkStrand:	524,288	1,028,403,333	0.067	# append calls = 3870
LinkStrand:	1,048,576	2,042,900,613	0.062	# append calls = 3870
LinkStrand:	2,097,152	4,071,895,173	0.065	# append calls = 3870
LinkStrand:	4,194,304	8,129,884,293	0.064	# append calls = 3870
LinkStrand:	8,388,608	16,245,862,533	0.074	# append calls = 3870
LinkStrand:	16,777,216	32,477,819,013	0.076	# append calls = 3870
LinkStrand:	33,554,432	64,941,731,973	0.060	# append calls = 3870
LinkStrand:	67,108,864	129,869,557,893	0.064	# append calls = 3870
LinkStrand:	134,217,728	259,725,209,733	0.064	# append calls = 3870
LinkStrand:	268,435,456	519,436,513,413	0.066	# append calls = 3870
LinkStrand:	536,870,912	1,038,859,120,773	0.066	# append calls = 3870

Average time is: 0.065090909

For ecoli_4x.txt

dna length = 18,556,884
cutting at enzyme gaattc

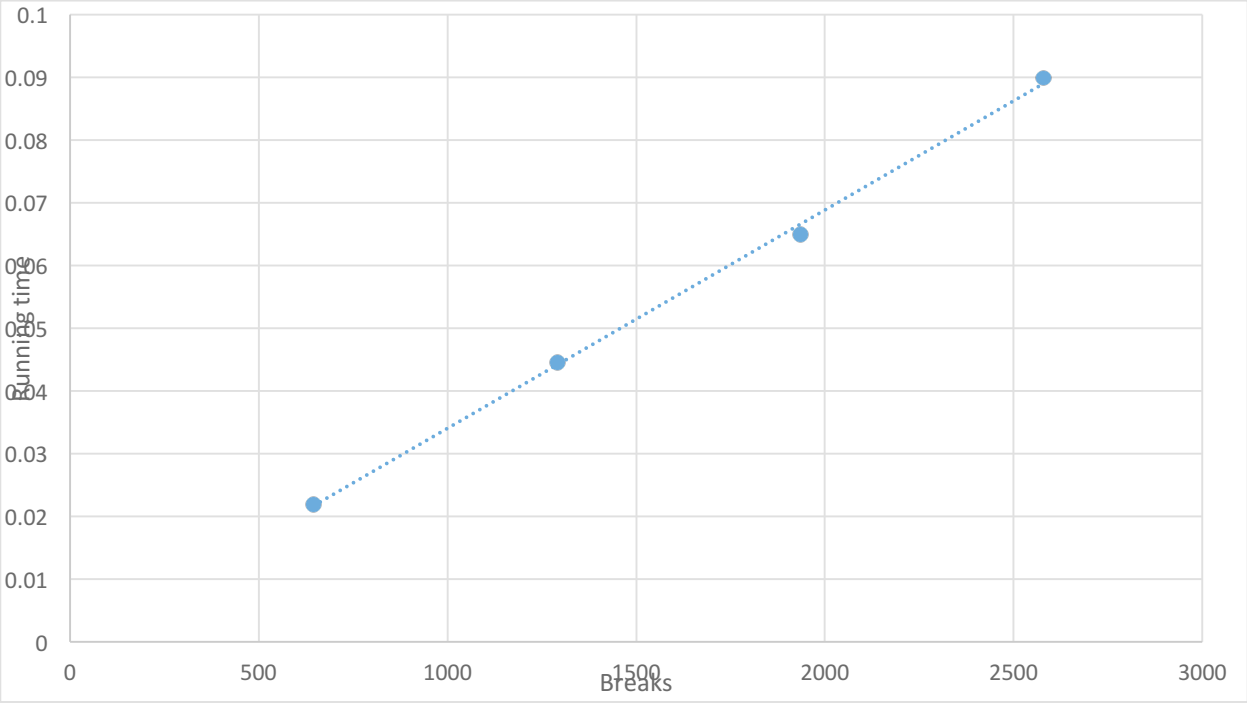
Class	splicee	recombtime		
LinkStrand:	256	19,201,884	0.081	# append calls = 5160
LinkStrand:	512	19,862,364	0.087	# append calls = 5160
LinkStrand:	1,024	21,183,324	0.081	# append calls = 5160

LinkStrand:	2,048	23,825,244	0.081	# append calls = 5160
LinkStrand:	4,096	29,109,084	0.084	# append calls = 5160
LinkStrand:	8,192	39,676,764	0.084	# append calls = 5160
LinkStrand:	16,384	60,812,124	0.098	# append calls = 5160
LinkStrand:	32,768	103,082,844	0.095	# append calls = 5160
LinkStrand:	65,536	187,624,284	0.092	# append calls = 5160
LinkStrand:	131,072	356,707,164	0.095	# append calls = 5160
LinkStrand:	262,144	694,872,924	0.098	# append calls = 5160
LinkStrand:	524,288	1,371,204,444	0.091	# append calls = 5160
LinkStrand:	1,048,576	2,723,867,484	0.147	# append calls = 5160
LinkStrand:	2,097,152	5,429,193,564	0.087	# append calls = 5160
LinkStrand:	4,194,304	10,839,845,724	0.079	# append calls = 5160
LinkStrand:	8,388,608	21,661,150,044	0.084	# append calls = 5160
LinkStrand:	16,777,216	43,303,758,684	0.089	# append calls = 5160
LinkStrand:	33,554,432	86,588,975,964	0.089	# append calls = 5160
LinkStrand:	67,108,864	173,159,410,524	0.090	# append calls = 5160
LinkStrand:	134,217,728	346,300,279,644	0.083	# append calls = 5160
LinkStrand:	268,435,456	692,582,017,884	0.085	# append calls = 5160
LinkStrand:	536,870,912	1,385,145,494,364	0.079	# append calls = 5160

Average time is: 0.089954545

ecoli_2x.txt is double the size of ecoli.txt.
ecoli_3x.txt is three times the size of ecoli.txt.
ecoli_4x.txt is four times the size of ecoli.txt.

The relationship between is that breaks=appends/2



Here a plot is generated for number of breaks and running time. As file size increases from 1 to 4, the number of breaks increases proportionally, and the running time increases as well. Also, the relation between the number of breaks and running time is almost perfectly linear. Thus, LinkStrand has O(B) runtime.

This is the case because the append for LinkedList has runtime O(1) and the append for StringBuilder has runtime O(N). And since number of breaks is number of append/2, we can use O(B) for measurement. So LinkStrand has O(B) runtime.

Extra Credit

To only reverse String once, a HashMap is used to store the reversed String for latter use.

```
public IDnaStrand reverse() {
		StringBuilder first=new StringBuilder(myFront.value);
		String temp=first.reverse().toString();
		HashMap<String,String> strand=new HashMap<String,String>();//use Hashmap to store
reversed String
		//this will only reverse String once
		strand.put(myBack.value, temp);
		LinkStrand tempo=new LinkStrand(temp);
		Node back=myFront.next;
		while(back!=null)
		{
			StringBuilder rest=new StringBuilder(back.value);
			String tempRest="";
			if(!strand.containsKey(back.value))
			{
				tempRest=rest.reverse().toString();
				strand.put(back.value, tempRest);
			}
			else
			{
				tempRest=strand.get(back.value);
			}
			Node front=new Node(tempRest,tempo.myFront);
			tempo.myFront=front;
			Node n=back.next;
			back=n;
		}
		tempo.mySize=mySize;
		tempo.myAppends=myAppends;
		this.myFront=tempo.myFront;
		this.myBack=tempo.myBack;
		return this;
	}
```