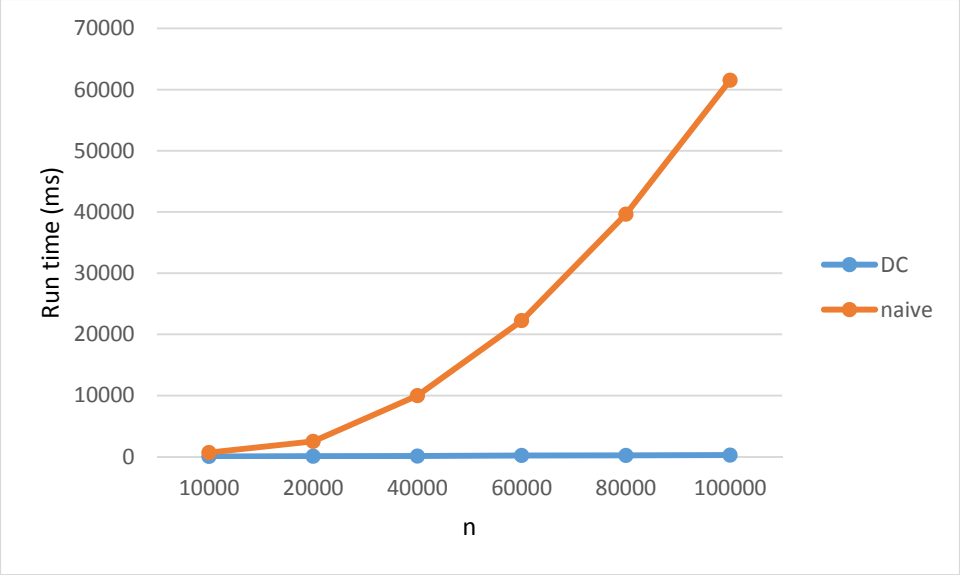
README.pdf
Josephine Lee
leejosephines@wustl.edu
CSE241 lab 1

Part 1: Description of implementation

For the naïve algorithm, I checked every pair of points using two nested for loops to find the closest pair. For the DC algorithm, I implemented it as described in class (and as requested, will not recapitulate it here). ifLeftOf() was particularly useful by allowing a linear pass-through of pointsByY[] to divide points according to whether they were in xL or xR while still retaining their sorting by Y.
I did not notice anything particularly interesting while implementing these algorithms. My implementation, to my knowledge, is not buggy.

Part 2: Comparison

Graph:



Data table:

n	10000	20000	40000	60000	80000	100000
DC	32	63	93	188	203	250
naïve	672	2500	9985	22235	39632	61530

For 100 randomly generated inputs of size 50000,
DC run time (milisec): average = 36.96, max = 188.0, min = 15.0, range = 103
Naive run time (milisec): average = 15710.57, max = 17685.0, min = 14600.0, range = 3086
I do not know how to compute a 95% confidence interval.

For 100 runs on the same input of size 50000,
DC run time (milisec): average = 40.43, max = 361.0, min = 29.0, range = 332
Naive run time (milisec): average = 15744.9, max = 17302.0, min = 14555.0, range = 2747

Variation in these 100 runs as reflected by the range must be due to external factors. Since this variation is similar (within the same order of magnitude) to that in the 100 randomly generated inputs of size 50000, it is likely that variation from one input to the next is small.

Part 3: Crossover

From the first comparison in part 2, the DC algorithm already seems to win out the naïve algorithm at n=10000. Thus, the crossover point (where the DC algorithm consistently wins) likely lies before this. To have a running time of at least 5 seconds (due to reasons as mentioned in the lab instructions), I have choosen k=10,000. I generated a new set of random points for every run.

In summary, I found that at around n>1875, the worst case DC begins to consistently win the best case naïve. At around n<30, naïve begins to become better than DC on average.

This is the data from all the tests I did:

n	results	comments
5000	DC run time (milisec): average = 2.4687, max = 61.0, min = 1.0 Naive run time (milisec): average = 149.4065, max = 360.0, min = 145.0	DC_{max} still < $naïve_{min}$, try a smaller n
2500	DC run time (milisec): average = 1.1927, max = 16.0, min = 0.0 Naive run time (milisec): average = 38.3638, max = 190.0, min = 36.0	DC_{max} still < $naïve_{min}$, try a smaller n
1250	DC run time (milisec): average = 0.5725, max = 11.0, min = 0.0 Naive run time (milisec): average = 9.6481, max = 42.0, min = 8.0	DC_{max} > $naïve_{min}$, try something higher
1875	DC run time (milisec): average = 0.8978, max = 19.0, min = 0.0 Naive run time (milisec): average = 22.1538, max = 78.0, min = 20.0	$DC_{max} \approx naïve_{min}$

At around n=1875, the worst case DC begins to consistently win the best case naïve.

I also explored some lower n values to find when DC_{avg} > $naive_{avg}$. The total run time for n=1250 was approaching 5sec, so to ensure a running time of at least 5 seconds, I increased k to k=100,000

n	results	comments
1000	DC run time (milisec): average = 0.42541, max = 14.0, min = 0.0 Naive run time (milisec): average = 6.3354, max = 37.0, min = 5.0	DC_{avg} still < $naïve_{avg}$, try a smaller n
500	DC run time (milisec): average = 0.19884, max = 10.0, min = 0.0 Naive run time (milisec): average = 1.54137, max = 18.0, min = 0.0	DC_{avg} still < $naïve_{avg}$, try a smaller n
	k was increased to k=500,000 for the following tests	
250	DC run time (milisec): average = 0.091892, max = 25.0, min = 0.0 Naive run time (milisec): average = 0.406214, max = 17.0, min = 0.0	DC_{avg} still < $naïve_{avg}$, try a smaller n
	k was increased to k=2,000,000 for the following tests	
125	DC run time (milisec): average = 0.036632, max = 25.0, min = 0.0 Naive run time (milisec): average = 0.1116255, max = 20.0, min = 0.0	DC_{avg} still < $naïve_{avg}$, try a smaller n
62	DC run time (milisec): average = 0.014921, max = 10.0, min = 0.0 Naive run time (milisec): average = 0.0255315, max = 24.0, min = 0.0	DC_{avg} still < $naïve_{avg}$, try a smaller n
30	DC run time (milisec): average = 0.0067355, max = 9.0, min = 0.0 Naive run time (milisec): average = 0.005776, max = 3.0, min = 0.0	$DC_{avg} \approx naïve_{avg}$

At around n<30, naïve begins to become better than DC on average