Maayez Imam 915342727

report.pdf

Using the dataset generator (create_dataset) and the working implementation of the program (vector_search_reference), I tested various sizes of datasets to compare the average running times. I ran each dataset three times and averaged the running times to make sure I got accurate and precise measurements. I started off with fairly small datasets (5 vectors and 5 magnitudes) as a base, but I eventually expanded to fairly large datasets (500 vectors and 500 magnitudes). My findings are displayed in the table below.

Search Method	# of Vectors	# of Magnitudes	3 CPU Times (microseconds)	Average CPU Time (microseconds)	Least Time Search Method	Matches
Linear	5	5	11.195 7.75 7.588	8.844	Binary	1
Binary	5	5	8.24 5.77 11.317	8.442		
Linear	5	10	8.22 10.407 7.127	8.585	Linear	2
Binary	5	10	14.8 9.455 12.653	12.303		
Linear	10	5	10.613 9.889 5.735	8.746	Binary	1
Binary	10	5	8.716 7.451 6.829	7.665		
Linear	20	20	13.918 22.564 17.348	17.943	Binary	8
Binary	20	20	11.396 12.668 15.452	13.172		
Linear	25	50	13.53 12.824 12.633	12.996	Linear	22
Binary	25	50	26.684 18.58 21.968	22.411		
Linear	50	25	24.475 21.669	32.067		

			50.056		Binary	8
Binary	50	25	12.448 43.576 27.264	27.763		
Linear	100	100	113.242 246.274 182.788	180.768	Binary	43
Binary	100	100	125.865 27.575 124.905	92.782		
Linear	100	500	677.865 348.14 869.116	631.707,	Binary	167
Binary	100	500	290.72 102.034 168.92	187.225		
Linear	500	100	642.179 568.652 1315.01	841.947	Binary	37
Binary	500	100	106.439 194.853 196.442	165.911		
Linear	500	500	1675.63 1651.94 1662.19	1663.253	Binary	180
Binary	500	500	159.727 161.175 161.397	160.766		

After analyzing my data, I can make some analysis regarding the choice between the two searching strategies. For smaller datasets, the choice between the better searching strategy varies. Both linear and binary are fairly even when it comes to running time. However, one thing I did notice is that linear always came out on top when there were more magnitudes to sort through than vectors. The reasoning for this could maybe be that the program searches through the vectors for matches to magnitudes, rather than searching through magnitudes for matches in magnitudes. This way, the dataset that has to be searched isn't as large as the dataset that has the values that need to be found, and thus linear search runs faster. Also, linear search came out on top when the number of matches was higher than the other datasets.

However, as the datasets start to get bigger, the CPU running times start to vary by quite a lot. The first time we really see this jump is when there are 100 vectors and 100 magnitudes. The running time for linear search is nearly double that of binary search. As the data sets get bigger from there, the gap between the running times between linear and binary continues to grow. So as the dataset gets bigger, binary search is more efficient.