CS 342

FALL 2019

PROJECT #1

PROJECT REPORT

BERK YILDIZ

21502040

I did my experiments with a computer which has processor intel core i5 vPro. The computer is seven years old and it is a slow machine when compared to today's computers. I used two different file sizes which have 500 integers and 1000 integers respectively.

Findings for filesize with 500 integers:

	k = 100	**k = 500**
N = 1	59ms	298ms
N = 2	134ms	579ms
N = 3	212ms	911ms
N = 4	309ms	1233ms
N = 5	347ms	1501ms

Table 1: Findings for filesize with 500 integers

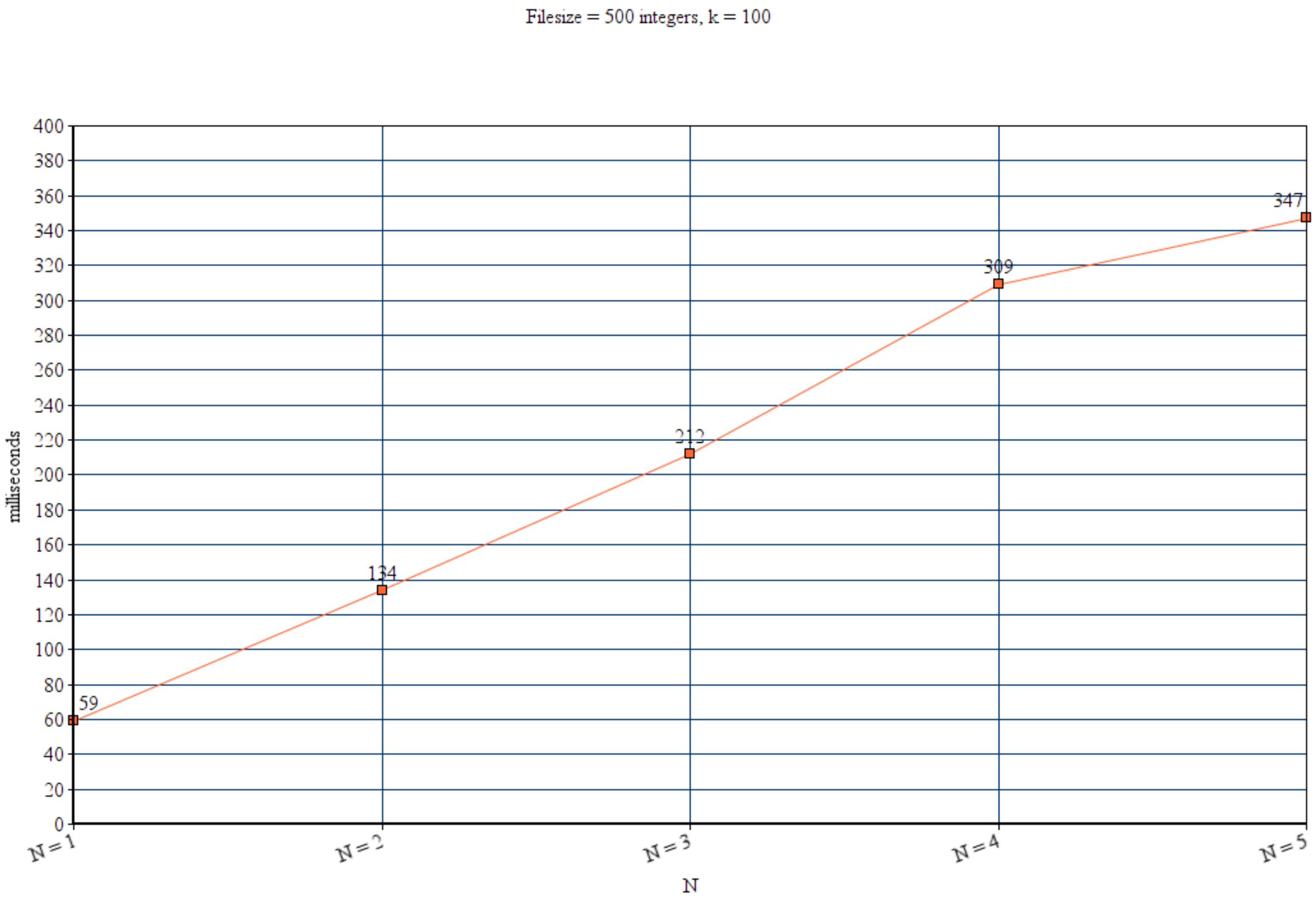

Table1: Findings for filesize=500 integers, k=100

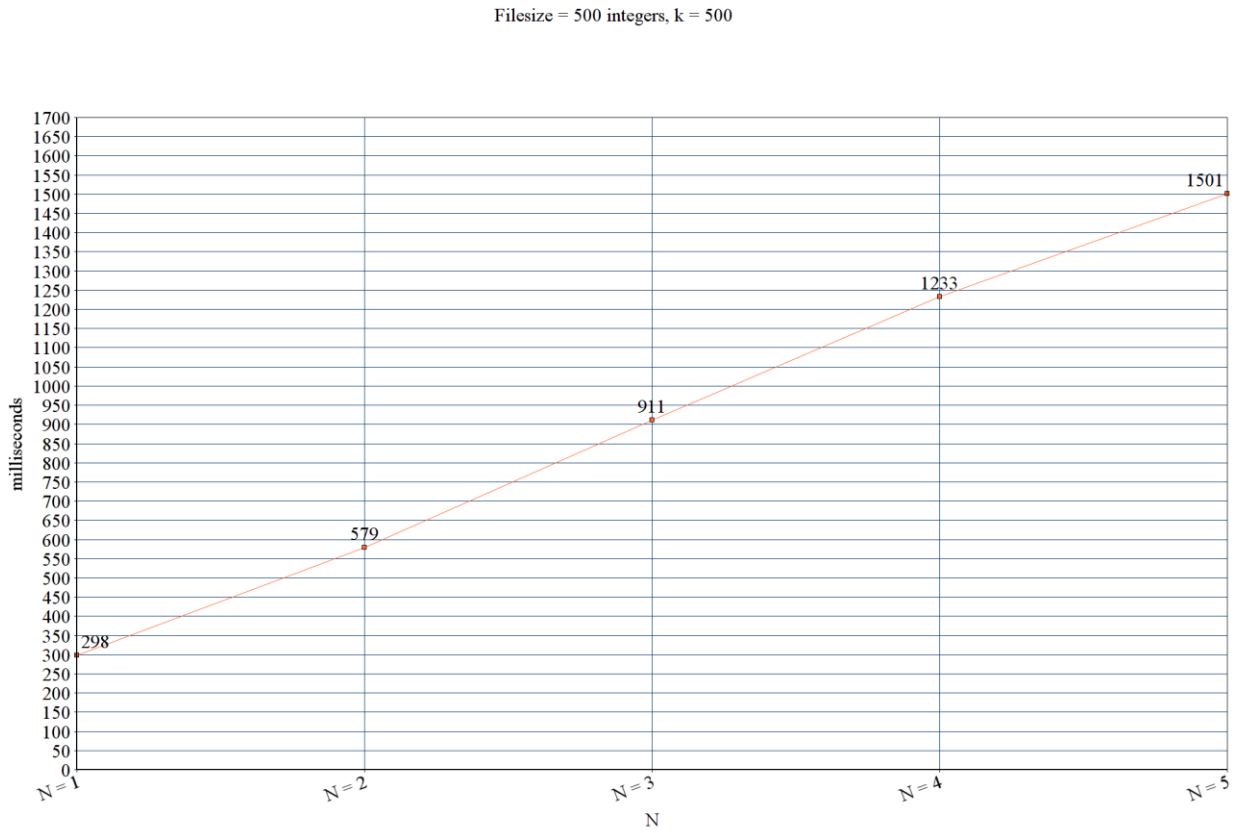

Table2: Findings for filesize=500 integers, k=500

Findings for filesize with 1000 integers:

	k = 100	k = 500	k=800
N = 1	71	323	510
N = 2	127	640	1012
N = 3	205	883	1497
N = 4	298	1167	1947
N = 5	356	1455	2543

Table 2: Findings for filesize with 1000 integers

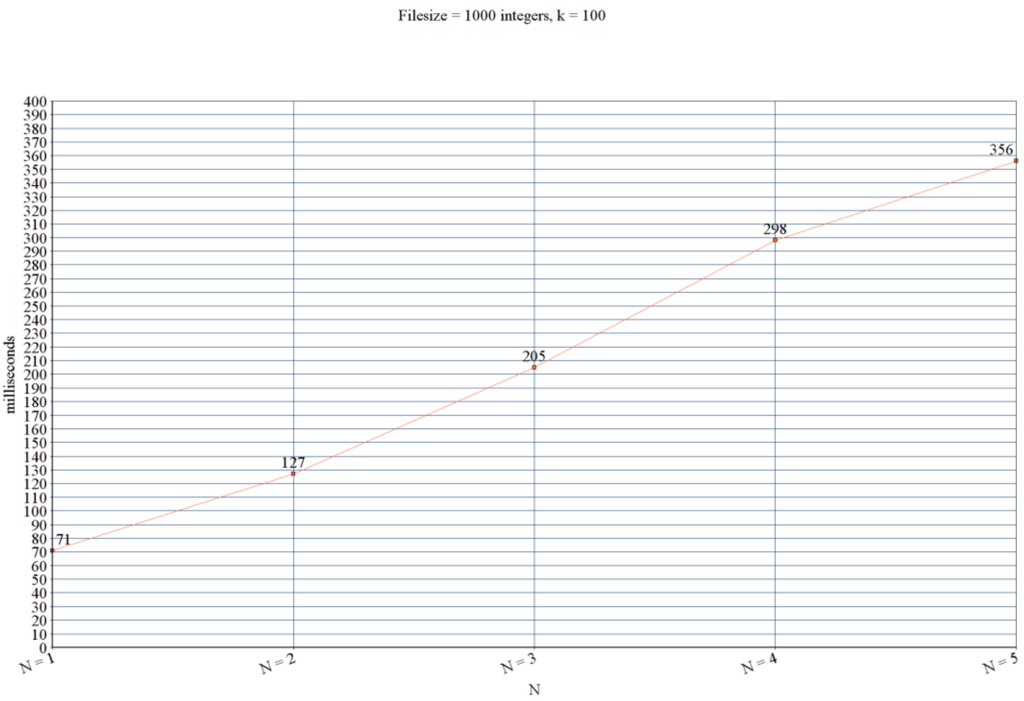

Table3: Findings for filesize=1000 integers, k=100

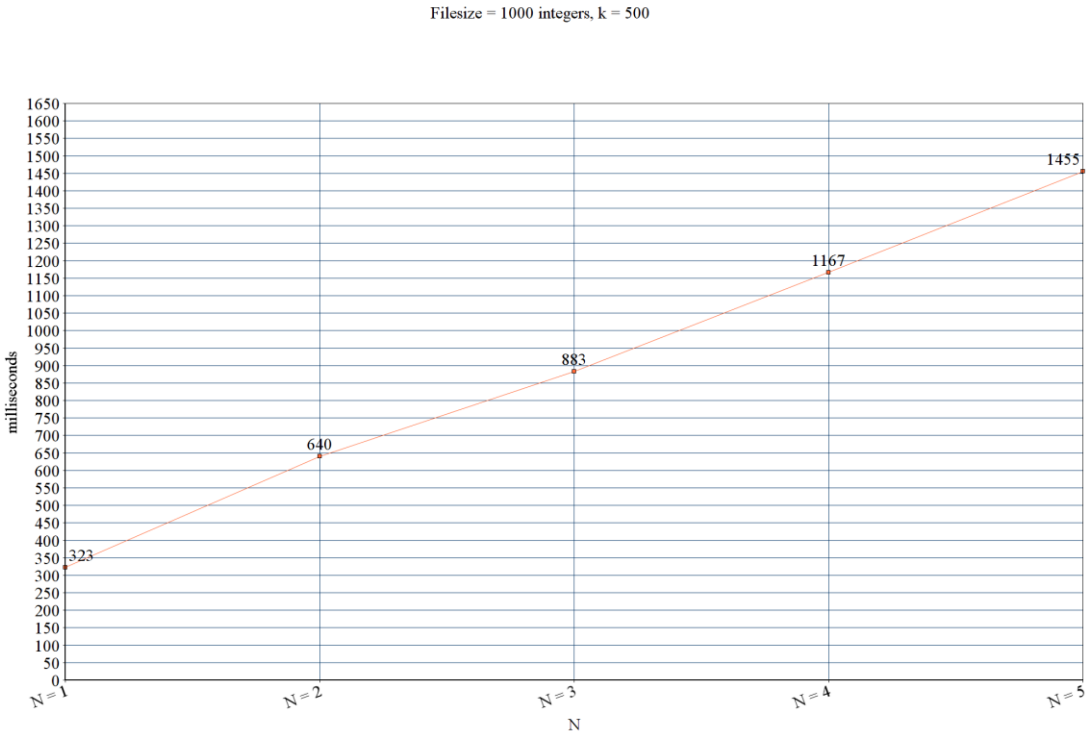

Table4: Findings for filesize=1000 integers, k=500

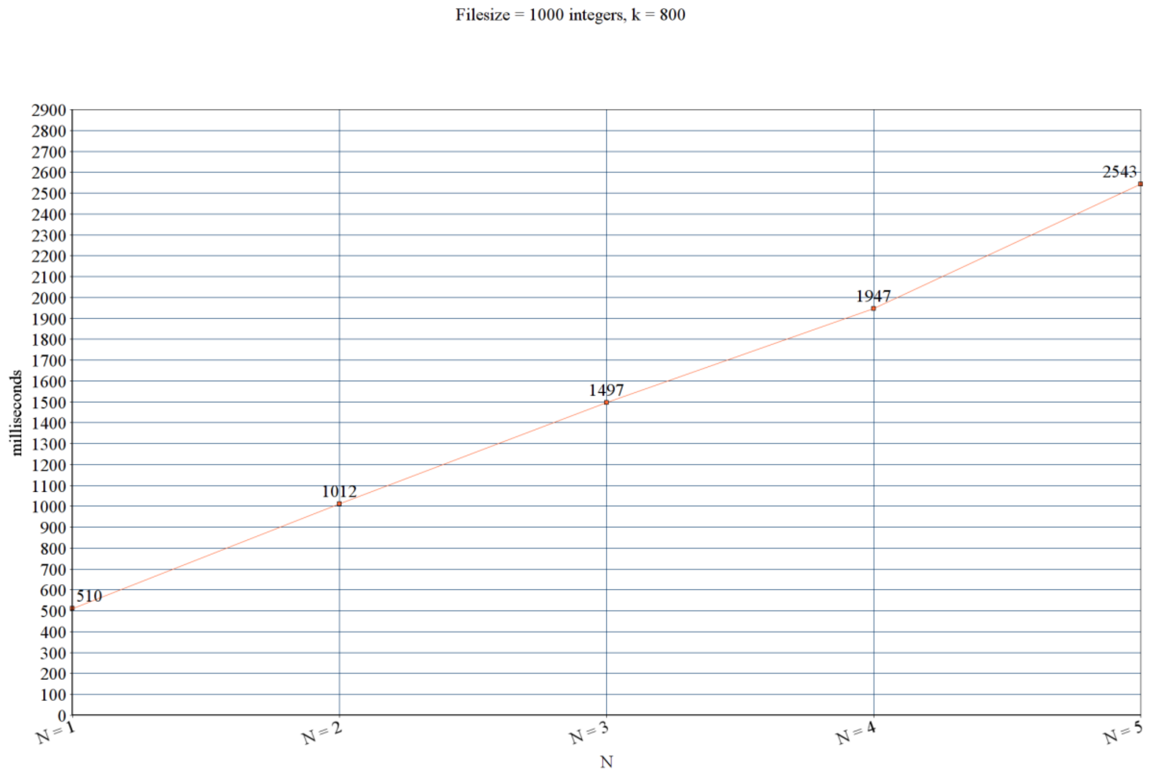

Table5: Findings for filesize=1000 integers, k=800

As the result of the experiment the dominant linearity in the graphs are visible. As the number of input files increases with constant k value, execution time increases directly proportional. The reason behind linearity is already CPU executes one file at a time and there is not any module which will reduce the execution time of CPU in the program and in my computer. So as the number of data increases, time for execution increases too. Also there is not an visible affect of filesize on execution time. Already opening a file is not a costly operation in our case and the number of inputs are pretty little values for a CPU process. So it is normal to not feel affect of filesize.