Introduction to Algorithms

Fall 2017

Workshop 4: SelectionSort

Make your graph as a Google doc, save it as Word or pdf, and add it to your branch.

Show your work in an online doc, and add it to your branch.

1. MovieSorter uses SelectionSort to rank movies by IMDb score. Fill in the missing code in two of the java files and compile.
2. The Data files have 20,000+ movies (including duplicates). One is in random order, one is sorted already, and one is in reverse order.
   1. [Graph](https://docs.google.com/document/d/1HQtv7VzbHILNazmqcMWuR7Gcc14PbYbvnGjDSNFMa68/edit?usp=sharing) the running times on each data file, for N=2000, 4000, 8000, 16000. Try each run three times and take the average time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Random | Try 1 | Try 2 | Try 3 | Average |
| N = 2000 | 0.029 | 0.025 | 0.024 | 0.026 |
| N = 4000 | 0.057 | 0.056 | 0.057 | 0.056 |
| N = 8000 | 0.203 | 0.216 | 0.215 | 0.211 |
| N = 16000 | 1.422 | 1.47 | 1.571 | 1.487 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reversed | Try 1 | Try 2 | Try 3 | Average |
| N = 2000 | 0.024 | 0.024 | 0.027 | 0.025 |
| N = 4000 | 0.068 | 0.066 | 0.067 | 0.067 |
| N = 8000 | 0.258 | 0.238 | 0.236 | 0.244 |
| N = 16000 | 1.608 | 1.412 | 1.425 | 1.481 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sorted | Try 1 | Try 2 | Try 3 | Average |
| N = 2000 | 0.022 | 0.027 | 0.032 | 0.027 |
| N = 4000 | 0.053 | 0.049 | 0.053 | 0.051 |
| N = 8000 | 0.163 | 0.17 | 0.163 | 0.165 |
| N = 16000 | 1.332 | 1.278 | 1.387 | 1.332 |

1. From your timing data analyze the rate of growth of SelectionSort using the doubling rule.

if the input doubles and the runtimes double, according to the doubling hypothesis, the runtime for this function is linear.

1. Analyze the SelectionSort [code](http://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/Selection.java.html): use number of array accesses as your cost model, and describe the rate of growth in tilde (~) notation.

Assuming that calling less() does not add to runtime the rate of growth for selection sort would be ~N2 because there are two array accesses in two nested for loops.

1. How does the order of the input data affect SelectionSort? Explain why, by referring to the [algorithm](https://drive.google.com/file/d/0B67e76GebdMFWDBnNl9FSm5kRVk/view?usp=sharing) or the [code](http://algs4.cs.princeton.edu/code/edu/princeton/cs/algs4/Selection.java.html). (How many times **less** and **exch** called?

In the selection sort code, less and exch are always called regardless of the input size because the if condition in the nested for loop is the only condition so it always calls within that loop and the exch() function is within the first for loop which runs from 0 to N.