2017/10/10

実践プログラミング　レポート

3J04岩崎 悠紀

・概要

　線形探索の一般の方、番兵版、二分探索のループ、再帰版の四つのプログラムを実行し、それぞれの実行速度を比較する。なお、比較する際には、それぞれ十回ずつ処理を回し、それぞれを10で割った平均の値を用いる。

・計算に使用した機器のスペック

　OS：OS X El Capitan ver.10.11.6

　CPU：Intel Core i7 2.8 GHz

　メモリ：8 GB 1333 MHz DDR3

・実行速度比較

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 種類 | 線形探索(通常) | 線形探索(番兵) | 二分探索(ループ) | 二分探索(再帰) |
|  | 2.343489 | 2.180207 | 0.016924 | 0.016577 |
|  | 2.347019 | 2.156968 | 0.018404 | 0.01616 |
|  | 2.299955 | 2.152971 | 0.015616 | 0.015726 |
|  | 2.322607 | 2.163072 | 0.017992 | 0.019176 |
|  | 2.317254 | 2.147598 | 0.016788 | 0.016432 |
|  | 2.275964 | 2.144638 | 0.015822 | 0.016101 |
|  | 2.227439 | 2.109972 | 0.020501 | 0.021267 |
|  | 2.328778 | 2.135807 | 0.018267 | 0.020267 |
|  | 2.317182 | 2.100932 | 0.015161 | 0.01767 |
|  | 2.358267 | 2.14814 | 0.015673 | 0.016033 |
| 実行時間（平均）[s] | 2.3137954 | 2.1440305 | 0.0171148 | 0.0175409 |

＊10回の処理はプログラムの中に組み込んでおり、その10回の平均を出力するようにしている。

＊データは、レコード数N=40000、キーの数は20000個としている。

・考察

　実行した結果わかったことは、実行速度は、

　線形探索(通常)<線形探索(番兵)<<二分探索(再帰)<二分探索(ループ)

、となっていた事だった。

　ではなぜ二分探索の方が線形探索よりもかなり早いのか。自分は、二分探索はまずクイックソートをしてから探索を始めるため、ソート済みのものを探索するため実行速度がかなり早かったのだと考えた。また、線形探索の番兵と一般の違いについてはおそらく1割程度のものだったと、実験の結果から考える。その1割については、ソースからわかるように、for文の中に条件文を入れるよりもwhile文で条件文をそのまま使う方が早いのだと考える。また、二分探索のループと再帰についてもほぼ同じだと思うが、再帰の方が処理が余分に出てしまうのでおそらく少しだけ時間がかかっているのだと思う。余分に出るというのは、再帰をしていると値が出たのにもかかわらず、returnしなくてはならないということであると思う。

　線形探索通常版と番兵版の違いについては、番兵版にすることでターゲットが最後に必ずあるため、いちいちfor文によって最後に達したかの判定の処理をしなくても良いので１割ほどの処理時間の差が出たのだと考えた。