PA1 report

Algs23s065 b09901036 邱士倫

Result of algorithm:

Case 1:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| case1 |  |  |  |  |  |  |  |  |
| input file | IS |  | MS |  | QS |  | HS |  |
|  | cpu time(ms) | memory(kb) | cpu time(ms) | memory(kb) | cpu time(ms) | memory(kb) | cpu time(ms) | memory(kb) |
| 1000 | 0.807 | 5904 | 0.558 | 5904 | 3.464 | 5904 | 0.306 | 5904 |
| 2000 | 2.517 | 5904 | 0.574 | 5904 | 4.917 | 5904 | 0.499 | 5904 |
| 4000 | 5.659 | 5904 | 1.152 | 5904 | 9.529 | 5904 | 0.96 | 5904 |
| 8000 | 16.275 | 6056 | 2.38 | 6056 | 14.614 | 6056 | 1.894 | 6056 |
| 16000 | 43.543 | 6056 | 3.481 | 6056 | 18.332 | 6056 | 2.424 | 6056 |
| 32000 | 165.06 | 6188 | 4.974 | 6188 | 35.36 | 6188 | 3.623 | 6188 |
| 1000000 | 177069 | 12144 | 152.745 | 14004 | 788.685 | 12144 | 138.696 | 12144 |

In case 1 (average case), the Merge sort and Heap sort have same tendency( O(nlgn) time) and Quick sort has same but with higher time cost due to randomized partition , which may increase time cost eventually. However, the Insertion sort should have distinctive tendency with its average case cost O(n^2) time.

Case 2:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| case2 |  |  |  |  |  |  |  |  |
| input file | IS |  | MS |  | QS |  | HS |  |
|  | cpu time(ms) | memory(kb) | cpu time(ms) | memory(kb) | cpu time(ms) | memory(kb) | cpu time(ms) | memory(kb) |
| 1000 | 0.132 | 5904 | 0.333 | 5904 | 2.549 | 5904 | 0.271 | 5904 |
| 2000 | 0.131 | 5904 | 0.438 | 5904 | 4.428 | 5904 | 0.462 | 5904 |
| 4000 | 0.136 | 5904 | 1.077 | 5904 | 7.715 | 5904 | 0.856 | 5904 |
| 8000 | 0.137 | 6056 | 1.222 | 6056 | 11.414 | 6056 | 1.156 | 6056 |
| 16000 | 0.141 | 6056 | 1.673 | 6056 | 15.252 | 6056 | 1.406 | 6056 |
| 32000 | 0.149 | 6188 | 3.203 | 6188 | 25.104 | 6188 | 2.563 | 6188 |
| 1000000 | 1.342 | 12144 | 65.936 | 14004 | 889.689 | 12144 | 75.562 | 12144 |

In case 2 (best case), which all data had been already sorted ,the Merge sort and Heap sort have same tendency (O(nlgn) time)and Quick sort has same but with higher time cost due to randomized partition , which may increase time cost eventually. However, the Insertion sort should have distinctive tendency(with lower time cost ) with its best case cost O(n) time instead of O(nlgn) time as Merge sort and Heap sort or even worse O(n^2) in some randomize partition of Quick sorts .

Case 3:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| case3 |  |  |  |  |  |  |  |  |
| input file | IS |  | MS |  | QS |  | HS |  |
|  | cpu time(ms) | memory(kb) | cpu time(ms) | memory(kb) | cpu time(ms) | memory(kb) | cpu time(ms) | memory(kb) |
| 1000 | 1.239 | 5904 | 0.349 | 5904 | 3.79 | 5904 | 0.279 | 5904 |
| 2000 | 3.795 | 5904 | 0.978 | 5904 | 5.987 | 5904 | 0.449 | 5904 |
| 4000 | 10.73 | 5904 | 1.04 | 5904 | 8.417 | 5904 | 0.574 | 5904 |
| 8000 | 26.33 | 6056 | 1.582 | 6056 | 12.588 | 6056 | 0.884 | 6056 |
| 16000 | 85.395 | 6056 | 1.937 | 6056 | 19.002 | 6056 | 1.572 | 6056 |
| 32000 | 331.053 | 6188 | 3.185 | 6188 | 34.125 | 6188 | 1.927 | 6188 |
| 1000000 | 361676 | 12144 | 73.438 | 14004 | 942.409 | 12144 | 74.288 | 12144 |
|  |  |  |  |  |  |  |  |  |

In case 3 (worst case), which all data had been already sorted backwardly ,the Merge sort and Heap sort have same tendency (O(nlgn) time) even compared with case 1 and case 2 . Quick sort has same but with higher time cost due to randomized partition , which may increase time cost eventually. However, the Insertion sort has much higher tendency of extremely high time cost ) with its worst case cost O(n^2) .

Conclusion

Among the four sorting algorithm have each method ways of using:

Quick sort may be a little inefficient compared to merge sort in every possible scenario ,but it takes less memory space than merge sorts. Heap sort is undoubtly fastest but with instable when more than two identical numbers. Insertion sort is an easy way to implement and sometimes efficient , however not so efficient in most scenarios. Merge sort can satisfy both stable and efficient but needs much more memory space when data is huge