

Thí nghiệm 1: Mảng tạo ngẫu nhiên (giới hạn random là $k = 10^8$)

Các thuật toán sort	10,000	20,000	40,000	60,000	80,000	100,000	120,000	140,000	160,000	200,000
Bubble Sort	3,569	13,338	41,779	93,480	114,435	168,957	237,289	508,347	643,475	1,099,560
Shaker Sort	2,798	8,915	28,081	65,536	91,268	148,042	207,294	454,226	558,035	877,799
Selection Sort	557	1,305	4,447	9,238	15,421	24,537	35,146	65,393	92,196	139,440
Insertion Sort	278	656	2,045	4,626	7,899	12,603	17,804	35,610	54,198	83,905
Binary Insertion Sort	224	517	1,578	5,513	6,086	9,613	13,692	25,434	38,489	56,040
Shell Sort	7	10	17	39	35	45	56	119	14	184
QuickSort	10	11	18	47	37	48	57	105	135	178
Qsort ©	18	25	42	103	87	11	132	228	316	372
HeapSort	19	26	43	104	89	116	145	99	345	397
Merge Sort	6	11	19	45	37	47	56	96	135	155
Counting Sort	1,566	1,437	1,120	1,662	1,376	1,098	1,092	1,547	1,882	1,604
Radix Sort	3	7	11	25	46	26	35	50	82	81
FlashSort	147	602	1,840	5,375	21	11	14	19	32	36

Các thuật toán sort	1,000,000	2,000,000	4,000,000	6,000,000	8,000,000	10,000,000	12,000,000	14,000,000	16,000,000	20,000,000
Shell Sort	1,564	3,184	6,538	9,306	15,051	12,659	27,072	31,974	35,948	39,831
QuickSort	1,186	2,002	3,933	7,560	7,377	6,715	15,751	18,530	18,162	23,833
Qsort ©	2,723	4,463	9,512	15,877	19,509	19,417	38,695	45,110	49,923	60,151
HeapSort	3,069	5,420	14,538	19,844	28,438	32,967	49,567	58,926	66,368	77,404
Merge Sort	942	1,599	4,591	5,821	9,208	8,340	14,061	16,652	17,714	24,651
Counting Sort	2,036	2,170	3,678	4,762	6,354	6,207	6,999	7,249	8,758	10,657
Radix Sort	376	717	1,688	2,521	4,302	4,209	6,102	6,457	8,848	11,261
FlashSort	314	904	1,571	2,329	3,896	3,974	5,484	4,760	7,748	10,405

Thí nghiệm 2: Mảng đã được xếp tăng dần (nửa mảng đầu có giá trị 0 -> n/2, nửa mảng sau có giá trị k-n/2 -> k)

Các thuật toán sort	10,000	20,000	40,000	60,000	80,000	100,000	120,000	140,000	160,000	200,000
Bubble Sort	587	2,257	9,418	20,767	24,961	52,113	60,890	70,478	86,951	124,950
Shaker Sort	0	3	0	0	3	0	0	0	1	1
Selection Sort	530	1,994	7,494	16,829	22,748	36,696	50,477	62,137	62,053	97,028
Insertion Sort	0	0	2	3	2	0	0	2	1	2
Binary Insertion Sort	3	5	10	17	18	15	30	42	25	34
Shell Sort	3	2	5	10	7	7	15	20	11	14
QuickSort	2	5	7	10	10	13	20	25	14	20
Qsort ©	12	25	43	87	77	80	150	203	117	149
HeapSort	8	15	25	45	40	43	75	100	59	75
Merge Sort	5	12	20	38	32	35	65	85	49	63
Counting Sort	2,050	2,408	1,901	2,215	1,420	1,200	1,822	2,015	1,063	2,129
Radix Sort	8	10	20	30	27	30	53	78	44	102
FlashSort	3	3	5	7	5	7	15	25	11	23

Các thuật toán sort	1,000,000	2,000,000	4,000,000	6,000,000	8,000,000	10,000,000	12,000,000	14,000,000	16,000,000	20,000,000
Shell Sort	174	370	785	1,210	1,595	1,963	2,293	2,761	3,297	4,120
QuickSort	230	467	966	1,596	1,637	2,335	2,628	2,913	3,393	4,399
Qsort ©	1,885	3,949	7,210	10,462	11,482	17,156	15,844	27,590	23,123	23,773
HeapSort	788	1,468	2,690	3,226	4,195	8,073	6,028	8,328	9,706	9,356
Merge Sort	582	1,207	2,267	2,410	3,163	4,348	4,543	5,313	7,745	7,749
Counting Sort	1,654	1,723	2,098	1,205	2,093	1,465	1,315	1,377	1,748	1,530
Radix Sort	385	755	1,613	1,634	3,970	2,811	3,198	3,738	4,876	5,391
FlashSort	95	192	397	410	947	688	820	962	1,080	1,343

Thí nghiệm 3: Mảng đã được xếp thứ tự ngược

Các thuật toán sort	10,000	20,000	40,000	60,000	80,000	100,000	120,000	140,000	160,000	200,000
Bubble Sort	4,757	17,070	68,164	113,468	212,526	322,052	450,943	608,860	832,171	1,153,180
Shaker Sort	4,684	16,687	52,295	118,264	215,575	310,859	414,350	637,455	824,793	1,446,830
Selection Sort	405	1,130	6,723	11,773	23,395	28,289	38,960	62,537	88,398	121,972
Insertion Sort	442	1,198	5,328	13,265	23,678	31,142	45,216	73,012	76,456	123,068
Binary Insertion Sort	335	927	3,723	8,646	13,242	22,496	32,886	60,829	70,450	82,600
Shell Sort	3	0	5	8	10	10	13	22	20	23
QuickSort	3	2	5	8	8	12	14	21	19	24
Qsort ©	10	15	32	48	57	73	93	146	137	151
HeapSort	8	8	17	25	30	40	46	73	64	76
Merge Sort	7	7	12	23	25	33	37	61	52	62
Counting Sort	1,627	1,220	1,215	1,981	1,110	1,080	1,100	1,478	1,156	1,066
Radix Sort	5	8	15	28	30	33	41	50	60	61
FlashSort	165	475	467	690	442	448	454	564	469	451

Các thuật toán sort	1,000,000	2,000,000	4,000,000	6,000,000	8,000,000	10,000,000	12,000,000	14,000,000	16,000,000	20,000,000
Shell Sort	268	614	1,287	1,947	2,939	3,260	3,898	3,195	5,583	5,852
QuickSort	267	580	1,249	1,672	1,984	2,924	2,726	2,184	3,823	4,318
Qsort ©	1,791	4,070	6,501	9,821	11,863	16,255	16,035	18,102	29,327	24,624
HeapSort	748	1,796	3,624	3,638	4,500	6,169	6,134	9,842	8,268	9,750
Merge Sort	556	1,298	3,140	2,812	3,504	4,317	8,089	7,094	6,872	8,093
Counting Sort	1,607	1,886	2,378	1,457	1,342	1,438	1,944	1,496	1,461	1,557
Radix Sort	340	1,006	1,892	2,037	2,500	3,327	4,216	4,387	5,059	6,296
FlashSort	519	832	961	792	794	895	973	1,097	1,244	1,486

Thí nghiệm 4: Mảng gần như có thứ tự

Các thuật toán sort	10,000	20,000	40,000	60,000	80,000	100,000	120,000	140,000	160,000	200,000
Bubble Sort	647	2,372	8,585	13,990	33,681	49,198	61,127	81,854	101,390	176,002
Shaker Sort	2	3	7	7	12	13	14	16	18	33
Selection Sort	556	1,745	6,604	12,353	28,417	29,599	39,344	65,759	101,824	140,706
Insertion Sort	0	0	1	1	2	2	4	4	5	5
Binary Insertion Sort	2	6	8	14	20	16	39	36	46	56
Shell Sort	2	2	6	8	10	9	22	21	27	32
QuickSort	2	3	6	8	11	9	23	22	28	32
Qsort ©	12	24	48	75	92	83	202	197	238	276
HeapSort	6	11	25	35	44	39	98	86	112	134
Merge Sort	5	11	24	29	37	32	77	72	93	100
Counting Sort	2,029	1,828	1,726	1,766	1,484	1,536	1,969	1,577	1,829	1,635
Radix Sort	4	8	16	28	32	69	64	51	62	83
FlashSort	1	2	6	5	8	19	17	12	15	20

Các thuật toán sort	1,000,000	2,000,000	4,000,000	6,000,000	8,000,000	10,000,000	12,000,000	14,000,000	16,000,000	20,000,000
Shell Sort	251	540	1,075	1,544	2,061	2,682	2,956	3,520	4,223	5,710
QuickSort	249	552	1,087	1,377	1,797	2,484	2,653	2,983	4,128	4,887
Qsort ©	2,368	4,363	8,203	13,531	12,782	22,140	20,415	21,488	29,886	34,517
HeapSort	1,013	1,598	3,047	5,296	4,786	7,654	7,060	8,069	9,235	15,529
Merge Sort	833	1,391	2,321	3,386	3,850	6,400	5,408	6,320	9,132	8,530
Counting Sort	2,444	2,155	1,834	1,771	1,398	2,052	1,427	1,552	2,199	1,652
Radix Sort	449	918	1,523	2,145	2,298	3,423	3,432	7,621	4,966	9,954
FlashSort	123	209	393	535	564	837	1,992	1,530	1,203	2,209

* Nhận xét:

Bubble Sort:

- Ưu điểm: Code đơn giản, dễ hiểu. Không tốn thêm mảng phụ.
- Nhược điểm:
 - + Nhưng độ phức tạp $O(N^2)$, không đủ nhanh nếu dữ liệu lớn.
 - + Chậm nhất trong các thuật toán sắp xếp cơ bản, do đổi chỗ hai phần tử gần nhau nên số lần đổi chỗ nhiều hơn.
 - + Số lượng phép so sánh không phụ thuộc vào tình trạng của dãy số ban đầu.
 - + Không nhận diện được tình trạng dãy đã có thứ tự hay có thứ tự từng phần.

Shaker Sort:

- Ưu điểm:
 - + Thuật toán sắp xếp cocktail là một dạng nâng cao của Bubble Sort nên nó có thể nhận biết được mảng đã được sắp xếp.
 - + Độ phức tạp cho trường hợp tốt nhất là $O(N)$.
- Nhược điểm:
 - Độ phức tạp cho trường hợp xấu nhất $O(N^2)$. Độ phức tạp trong trường hợp trung bình là $O(N^2)$.

Selection sort:

- Ưu điểm: Thuật toán đơn giản, cách làm dễ nên thường được sử dụng
- Nhược điểm:
 - + Số lần so sánh là $n(n-1)/2$, số lần hoán vị phụ thuộc vào tình trạng ban đầu của dãy số.
 - + Về thời gian tính toán thì $O(N^2)$ khá là chậm.
 - + Đây không phải là sort ổn định.

Insertion Sort:

- Ưu điểm: Với mảng gần như đã sắp xếp thì Insertion Sort là nhanh nhất.
- Nhược điểm:
 - + Độ phức tạp $O(N^2)$, không đủ nhanh nếu dữ liệu lớn.
 - + Giải thuật thực hiện tất cả $N-1$ vòng lặp while, do số lượng phép so sánh và dời chỗ này phụ thuộc vào tình trạng của dãy số ban đầu, nên chỉ có thể ước lượng trong từng trường hợp như sau: tốt nhất $2(n-1)$ phép gán, $(n-1)$ phép so sánh; xấu nhất $n(n+1)/2 - 1$ phép gán, $n(n-1)/2$ phép so sánh

Binary Insertion Sort: Làm giảm số lần so sánh, không làm giảm số lần dời chỗ. Ngoài ra, có thể cải tiến giải thuật chèn trực tiếp với phần tử cần chèn để giảm điều kiện kiểm tra khi xác định vị trí pos.

Shell Sort:

- Ưu điểm: đem lại hiệu quả cao hơn giải thuật Insertion Sort, tránh được các trường hợp phải trao đổi vị trí của hai phần tử xa nhau trong giải thuật sắp xếp chọn. Xử lý tốt trong trường hợp mảng gần như đã sắp xếp
- Nhược điểm: độ phức tạp trong trường hợp xấu nhất có thể lên đến $O(N^2)$

Quick Sort:

- Ưu điểm: chạy nhanh (nhanh nhất trong các thuật toán sắp xếp dựa trên việc so sánh các phần tử).
- Nhược điểm:
 - + Tùy thuộc cách chia 2 phần, nếu chia không tốt, độ phức tạp trong trường hợp xấu nhất có thể là $O(N^2)$
 - + Thuật toán không ổn định.
 - + Không hiệu quả trên những dãy đã được sắp xếp sẵn. Khi đó ta phải mất N lần gọi đệ quy và mỗi lần chỉ loại được 1 phần tử.

Quick Sort trong C: Dùng Intro sort (kết hợp giữa QuickSort và Insertion Sort) nên sẽ chậm hơn Quick Sort bình thường

Heap Sort:

- Ưu điểm:
 - + Chạy nhanh, độ phức tạp $O(N\log N)$.
 - + Không phải dùng thêm mảng phụ trong quá trình so sánh
- Nhược điểm:
 - + Không ổn định
 - + Trên thực tế heap sort không nhanh hơn quick .

Merge Sort:

- Ưu điểm:
 - + Chạy nhanh, độ phức tạp $O(N\log N)$.
 - + Với dữ liệu lớn và cần ít thao tác sắp xếp thì sẽ tối ưu hơn Quick sort.
- Nhược điểm:
 - + Cần dùng thêm bộ nhớ để lưu mảng.
 - + Code khó cài đặt.

Counting Sort:

- Ưu điểm:
 - + Xử lý tuyến tính
 - + Có thể xử lý hiệu quả với giá trị phần tử nhỏ
- Nhược điểm:
 - + Tốc độ phụ thuộc nhiều vào giá trị lớn nhất của mảng
 - + Tốn nhiều dung lượng, nếu mảng ít phần tử nhưng phần tử có giá trị rất lớn thì thuật toán vẫn xử lý rất chậm.
 - + Thuật toán chỉ dùng được cho số nguyên

Radix Sort:

- Ưu điểm: Có thể chạy nhanh hơn các thuật toán sử dụng so sánh. Trong trường hợp tổng quát, độ phức tạp là $O(N \log(\max(a)))$
- Nhược điểm: Không thể sắp xếp số thực

Flash Sort: Trong trường hợp dữ liệu đầu vào cân bằng với kích thước mỗi phân lớp xấp xỉ nhau, hiệu quả về thời gian của Flashsort sẽ đạt tốt nhất. Một điều hết sức thú vị là mặc dù đạt được hiệu quả tốt về thời gian nhưng Flashsort chỉ cần đến một không gian nhớ phụ rất khiêm tốn, thông thường nó yêu cầu ít hơn $0.1n$ bộ nhớ phụ để sắp xếp phần tử.