Given 11 blanks initially on which to perform a straight insertion sort [IS], the average number of moves (and compares) for each i-th item follows from Sedgewick and Flajolet [1].

i-Item	Average number of moves (or compares) at circa (n^2)/4 from [1; p 337]
1	0.25
2	1.00
3	2.25
4	4.00
5	6.25
6	9.00
7	12.25
8	16.00
9	20.25
10	25.00
11	30.25

The total number of moves [m] is:

m = ((((b ^ 2) * (int((n - 1) / b)) + ((mod (n - 1, b) + 1) ^ 2)) / 4) + (b * (int((n - 1) / b)))) .

The number of compares [c] is m plus the number of compares with the initial binary search:

c = (int(1 + log2(1 + (b * (0.5 * (int((n - 1) / b)) * (int((n - 1) / b) + 1)))))) + ((((b ^ 2) * (int((n - 1) / b)) + ((mod (n - 1, b) + 1) ^ 2)) / 4) + (b * (int((n - 1) / b)))) .

Performance

The average performance of compares for heap sort and quick sort is n lg n + n.

n	n lg n + n	BSAM compares	BSAM moves	Times fewer (compares)
100	764	381	372	2.01
1 000	10 966	3 754	3 738	2.92
10 000	142 877	37 520	37 497	3.81
100 000	1 760 964	375 017	374 988	4.70
1 000 000	20 931 569	3 750 033	3 749 997	5.58
10 000 000	2.4253497e+8	37 500 030	37 499 987	6.47
100 000 000	2.7575425e+9	3.75e+8	3.75e+8	7.35
1 000 000 000	3.0897353e+10	3.75e+9	3.75e+9	8.24

Hence for n from 100 to 1 000 000 000, BSAM performs from 2- to 8-times fewer compares and moves than do sorts with n lg n + n averages.

[1] Sedgewick, Robert and Flajolet, Phillipe. An Introduction to the Analysis of Algorithms. Addison-Wesley. 1996.