

Sorting and searching costs; the $\log_2 n$ miracle

Lesson 86 · Module 7 — Algorithms, Complexity & Epilogue · Discrete Mathematics · daily-maths.uk/discrete

Practice problems

1. A sorted list contains $n = 8192$ elements. What is the maximum number of probes required to find an element using binary search?
2. Compare the worst-case number of operations for a linear search versus a binary search on a sorted list of $n = 1,048,576$ elements.
3. If a comparison-based sorting algorithm is used on a list of $n = 100$ elements, what is the approximate minimum number of comparisons required in the worst case? (Use $\log_2 100 \approx 6.64$)

Solutions

1. A sorted list contains $n = 8192$ elements. What is the maximum number of probes required to find an element using binary search?

- The maximum number of probes is given by $\lceil \log_2 n \rceil$.
- We check powers of two: $2^{10} = 1024$, $2^{11} = 2048$, $2^{12} = 4096$, $2^{13} = 8192$.
- Since $8192 = 2^{13}$, the logarithm is exactly 13.

Answer: 13

2. Compare the worst-case number of operations for a linear search versus a binary search on a sorted list of $n = 1,048,576$ elements.

- For linear search, the worst case is n operations, which is 1,048,576.
- For binary search, the worst case is $\log_2 n$.
- Since $2^{20} = 1,048,576$, the binary search takes $\log_2(2^{20}) = 20$ operations.

Answer: Linear: 1,048,576; Binary: 20

3. If a comparison-based sorting algorithm is used on a list of $n = 100$ elements, what is the approximate minimum number of comparisons required in the worst case? (Use $\log_2 100 \approx 6.64$)

- The lower bound for comparison sorting is proportional to $n \log_2 n$.
- Substitute $n = 100$: $100 \times \log_2 100$.
- Using the approximation: $100 \times 6.64 = 664$.

Answer: Approximately 664