figure 5.1

Running times for small inputs

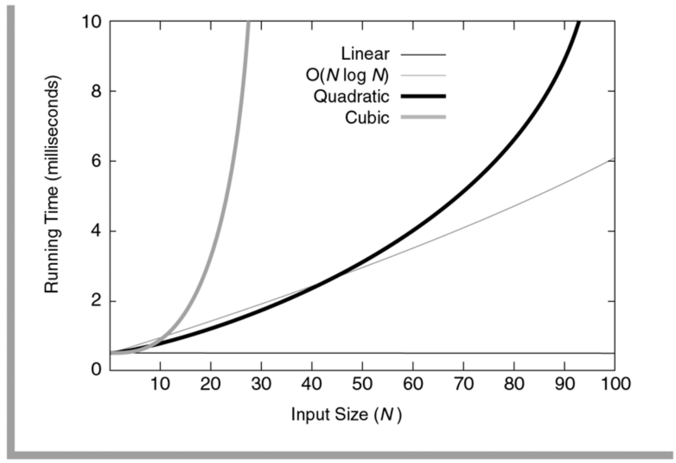
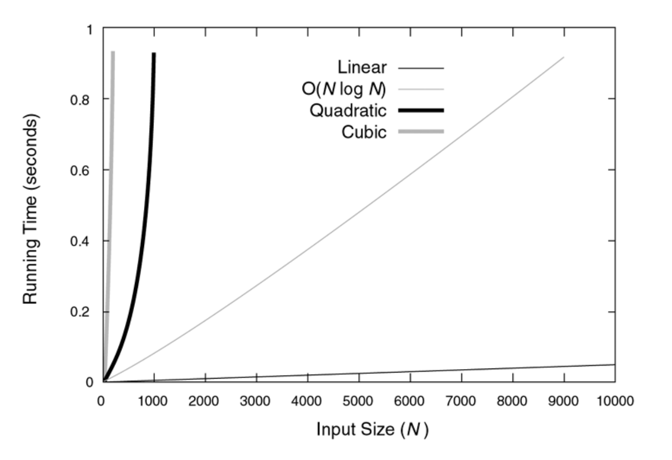


figure 5.2

Running times for moderate inputs



Function	Name
c	Constant
$\log N$	Logarithmic
$\log^2 N$	Log-squared
N	Linear
$N \log N$	$N \log N$
N^2	Quadratic
N^3	Cubic
2^N	Exponential

figure 5.3

Functions in order of increasing growth rate

Basic Principles of Analysis of Algorithms

- Determine which statements or expressions to count.
- For any n, one may determine:
  - Best case
  - Average case
  - Worst case
- If the best and worst case are in the same complexity class, so is the average case.
- Average case analysis is typically the hardest. It requires a probabilistic analysis.
- We are interested in the worst case behavior, when it comes to process control.
- We are interested in the average case behavior, when it comes to minimizing hardware cost for software that is run many times.
- We might perform a best case analysis, if we want to determine the average case and suspect that the best and worst case are the same.

Linear Search

```
public static int linearSearch(int[]a, int e){
  for (int i = 0; i < a.length; i++){
      if (a[i] == e) return i;
  }
  return -1;
}
```


Best Case Analysis of Linear Search

- Size of array is of length *n*.
- In **best** case, the element we are looking for is in the first position of the array.
- In this case, we have one comparison.
- O(1)

Worst Case Analysis of Linear Search

- Size of array is of length *n*.
- In **worst** case, the element we are looking for is in the last position of the array or not located in the array
- In these cases, we have to look at all elements of the array, giving n comparison.
- O(n)

1-7

Average Case Analysis of Linear Search

- Chances of looking for 1st element in array: 1/n
- Same for all other elements
- Number of elements to compare:
  - 1st element: 1
  - 2nd element: 2
  - nth element: n

Average Case Analysis of Linear Search

- Sum of all cases: 1/n*1 + 1/n*2 + … + 1/n*n
- Factor out 1/n: 1/n*(1 + 2 + … + n)
- Change notation: 1/n * $\sum_{i=0}^{n} i$
- By induction, you can show that:
  $\sum_{i=0}^{n} i$ = n*(n+1)/2
- Dividing by n: (n+1)/2
- O(n)