CS472 Module 3 Part A - Brute Force - Sometimes only a hammer will do

Athens State University

2013-11-18 Mon

Outline

Contents

1 What do you mean by "brute force"?

Brute force

Brute Force:

- A method of accomplishing something primarily by means of strength, without the use of mechanical aids or, in combat, without strategic planning or tactics.
- (computer science) A method of computation wherein all permutations of a problem are tried manually until one is found that provides a solution, in contrast to the implementation of a more intelligent algorithm.

Brute Force

Many problems can be solved by brute force

- But the consequences may be many

Examples

- Searching for a key of a given value in a list
- Computing $n!$
- Computing a^n

2 Examples of brute force algorithms

Example: Algorithms for Numerical Linear Algebra

- Many of the algorithms used for matrix operations fit into this class
- Consider matrix multiplication

  The m by p product of two matrices $C = A * B$ where A is a m by n matrix and B is a n by p matrix can be computed using the formula:

$$c_{ij} = \sum_{r=1}^{n} a_{i,r} b_{r,j}$$

Can we write an algorithm to compute this product and what is the rate of growth of the algorithm?

Example: Brute-Force String Matching

pattern a string m characters for which we are searching

text a (longer) string of n characters in which we are searching

Problem find a substring in the text that matches the pattern

Algorithm Design

1. Align pattern at start of text.
2. Moving from left to right, compare each character of the pattern to corresponding character in the text until
   - all characters are found to match (success)
   - a mismatch if detected (failure)
3. While pattern is not found and rest of text not exhausted, realign pattern one position to right and repeat step 2

Example: Brute-Force String Matching

Bit patterns

- Pattern: 0010011
- Text: 10010101101100110010111010

String matching

- Pattern: happy
- Text: It is never too late to have a happy childhood.

Example: Brute-Force String Matching

```
Algorithm 1: Brute Force String Matching
Input: An array T[0..n-1] of n characters and an array p[0..m-1] of m characters
Output: The index of the first character in the text starting a matching substring or -1 if search fails
for $i \leftarrow$ 0..(n-m) do
    j ← 0;
    while (j < m) and (P[j] = T[i+j]) do
        j ← j+1 ;
    if $j = m$ then
        return i ;
return -1;
```

Example: Brute Force Polynomial Evaluation: Problem

Find the value of the polynomial

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

at a point $x = x_0$

Example: Brute Force Polynomial Evaluation: Algorithm

```
Algorithm 2: Brute Force Polynomial Evaluation
Input: An array containing the polynomial coefficients
Output: The value of p(x) at point
p ← 0.0 ;
for $i \leftarrow$ [n..0] do
    power ← 1;
    for $j \leftarrow$ [1..i] do
        power ← power * x;
    p ← p + a[i] * power;
return p;
```

Example: Brute Force Polynomial Evaluation: Analysis

- Notice that the innermost loop of the algorithm is computing x^n
  - And using the brute force method
  - So, can count the number of multiplications

$$M(n) = \sum_{i=0}^{n} \sum_{j=1}^{n} 1 = \sum_{i=0}^{n} i = \frac{n(n+1)}{2} \in O(n^2)$$

Example: Brute Force Polynomial Evaluation: An improvement?

- The original algorithm computes powers of x without taking into account any relationship between them
- So, suppose we compute from highest term to lowest term
  - Better, but means we have to use division (why? and why bad?)
- Suppose we compute x^i by using x^{i-1}?

Example: Brute Force Polynomial Evaluation: an improvement?

Algorithm 3: Brute Force Polynomial Evaluation, Version 2

```
Input: An array containing the polynomial coefficients
Output: The value of p(x) at point
p ← a[0] ;
power ← 1;
for i ← 1..n do
    power ← power * x;
    p ← p + a[i] * power;
return p;
```

Example: Brute Force Polynomial Evaluation: improvement analysis

Again, we count the number of multplications:

$$M(n) = \sum_{i=1}^{n} 2 = 2n$$

Victory! We've gone from an $O(n^2)$ algorithm to $O(n)$ algorithm

Can we do better? No, because for any polynomial of degree n at a point x, we have to process $n+1$ coefficients.

3 Some observations about brute force algorithms

Man or Meta-Man?

- Well... actually heuristic vs meta-heuristic

  Heuristic A technique for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution

  Meta-heuristic A higher-level procedure or heuristic that finds, generates, or selects a lower-level procedure or heuristic for solving a problem

- One can brute force as an example of a meta-heuristic

What can we say about brute force algorithms?

- Brute force algorithms have many positive characteristics
  - Simple to implement
  - Will always find a solution if one exists
- Brute force algorithms have many more negative characteristics
  - Cost is proportional to number of candidate solutions
  - This number tends to grow quickly as size of problem increases
- Result is that such algorithms are used
  - When problem size is limited
  - When we can find problem-specific heuristics to cut back the number of solutions
  - When simplicity of implementation is more important than speed
- Brute force algorithms oft used as a baseline comparison