

Algorithms: Brute Force

Model 1: Some Problems and Algorithms

Consider the following scenarios:

- (a) You are given a list of 64-bit integers, and asked to sort them in increasing order.
- (b) You are given a collection of 2D points, and asked to find the two points which are closest to each other.
- (c) You are given a set of (positive and negative) integers, and asked to find a subset which sums to zero (if there is one).
- (d) You are given two positive integers a and b and asked to find their greatest common divisor, that is, the largest positive integer which evenly divides both a and b .
- (e) You are asked whether there is any way to visit all 48 contiguous US states by car without entering the same state twice.

Here are some algorithms intended to solve the above scenarios:

- (a) List every possible ordering of the given integers. Test each one to see whether it is sorted, and stop as soon as a sorted ordering is found.
- (b) List each possible pair of points, and compute the distance between each. Return the points with the smallest distance.
- (c) List every subset of the given set, and find the sum of each. If any subset is found with a sum of zero, return it.
- (d) (*You will fill in this algorithm in Question 4.*)
- (e) List every possible ordering of the 48 states. Check each ordering to see whether every pair of adjacent states in the ordering is also geographically adjacent. If such an ordering is found, output “yes”; else output “no”.

1 For each scenario, identify the *input(s)* to the problem.

(a)

(b)

(c)

(d)

(e)

2 For each scenario, identify the desired *output(s)*.

(a)

(b)

(c)

(d)

(e)

Learning objective: Students will analyze problems in terms of inputs and outputs.

Learning objective: Students will write brute-force algorithms to solve search problems.



- 3 All the algorithms in Model 1 are called *brute force algorithms*. Using one or more **complete English sentences**, write down a definition of what a *brute force algorithm* is.

- 4 Fill in this description of a brute force algorithm for scenario (d):

List all the _____

and output _____.

- 5 For which of the scenarios (a)–(e) do you think the given algorithm is the fastest possible algorithm? For which ones do you think a faster algorithm is possible?

I don't expect you to know the right answer for all of these!



Brute-Force Algorithms (15m)

Report out on definitions of brute force.

Closure (5m)

A lot of this semester is going to focus on approaches and techniques for doing better than brute-force. But we will also explore limits: for example, for scenario (c) we strongly suspect that it's not possible to do fundamentally better than brute force!

