Lower Bounds

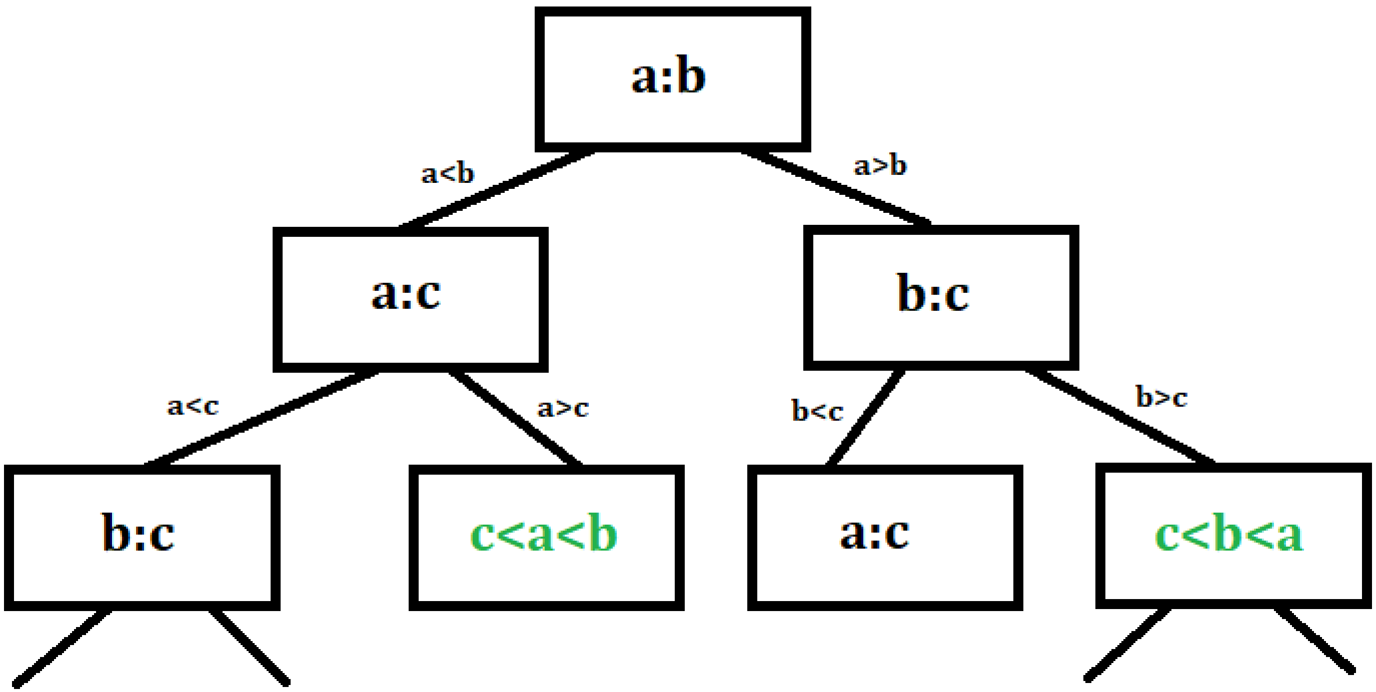
Sorting

Problem: Sort n unsorted (distinct) numbers.
Example: $7\ 2\ 1\ 4\ 9\ 6\ 3\ 8\ 5 \to 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9$
Lots of algorithms: $O(n \log n)$
Lower bound: $\Omega(n)$
Recall: $\log(n!) = \Theta(n \log n)$
Recall: In any binary tree T, $height(T) \geq \log_2 |leaves(T)|$
In the abstract, we need to create an algorithm that takes in a list of variables a, b, c, ... and produces their sorted order.
One unit of work will be a comparison: comparing one variable to another.

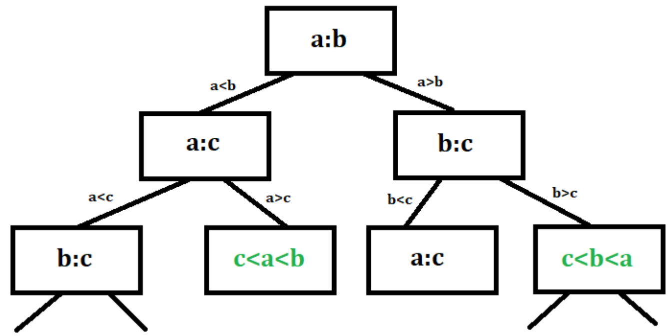
Decision Trees

Any "reasonable" algorithm can be represented by a decision tree.

Assume that we are trying to sort a, b, c.



Decision Trees: Reasoning



What does the height of the decision tree represent?

The worst case number of comparisons for the algorithm

At least how many leaves must the decision tree have?

$n!$ (one for each possible permutation of n numbers)

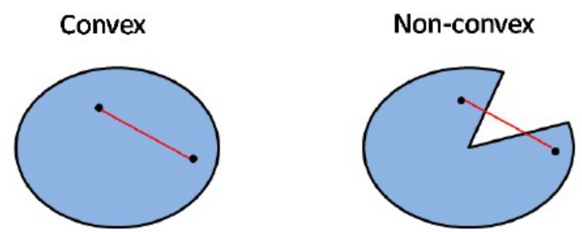
Let $|A(n)| \equiv$ worst case number of comparisons for algorithm A to sort n numbers.

$$|A(n)| = h(T) \geq \log_2 |leaves(T)| \geq \log_2(n!) \sim n \log n$$

The sorting problem is $\Omega(n \log n)$!

Convexity

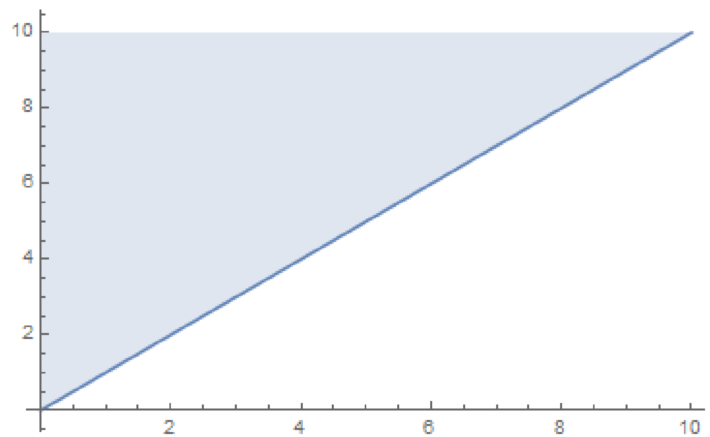
A space is convex if a straight line between any two points in the space remains entirely within the space.

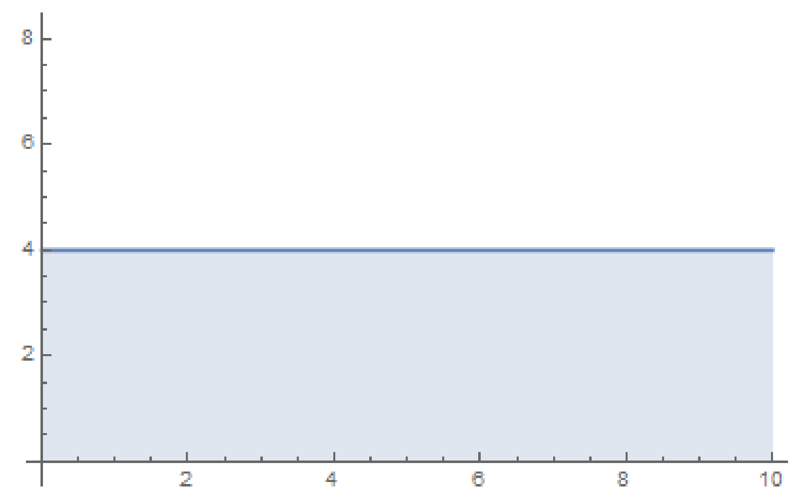


Observation: In n-dimensional space (where each point is of the form $(x_1, x_2, \ldots, x_n)$), the following half spaces are convex:

$\forall i \neq j : x_i < x_j, x_i \leq x_j, x_i > x_j, x_i \geq x_j$

$\forall a : x_i < a, x_i \leq a, x_i > a, x_i \geq a$

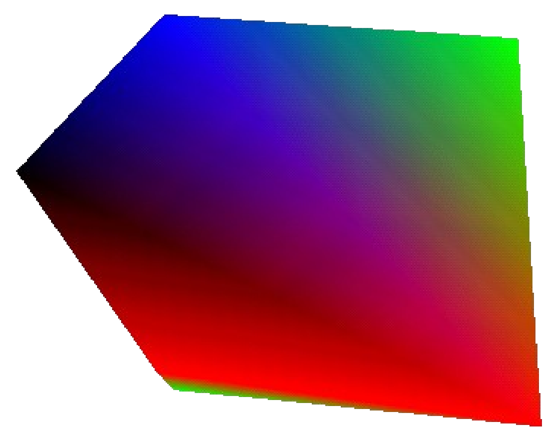




A Convexity Fact

Claim: The intersection of any collection of convex sets is convex.

Proof: Consider two points x and y that are in every set of the collection (and hence the intersection). By definition, because every set is convex, every point on the line that connects x and y is also in each set (and hence the intersection).



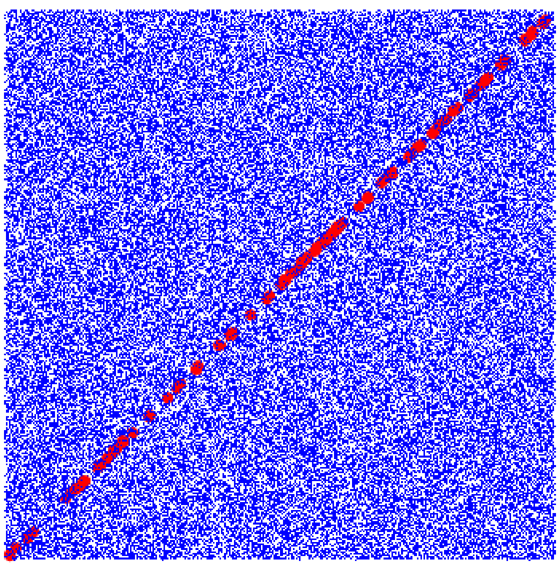
What would the intersection of multiple half-spaces look like?

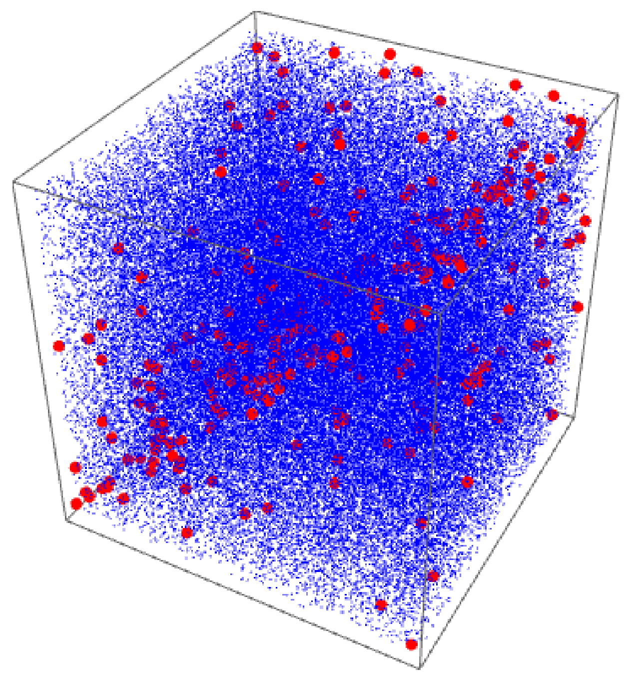
Element Uniqueness

Assume that you are given a set of n numbers $(x_1, x_2, \ldots, x_n)$ and the problem is to determine whether any two of them are equal.

One basic operation is a *comparison*.

Each possible input $(x_1, \ldots, x_n)$ can be thought of as a point in n-dimensional space labeled with a YES or NO.



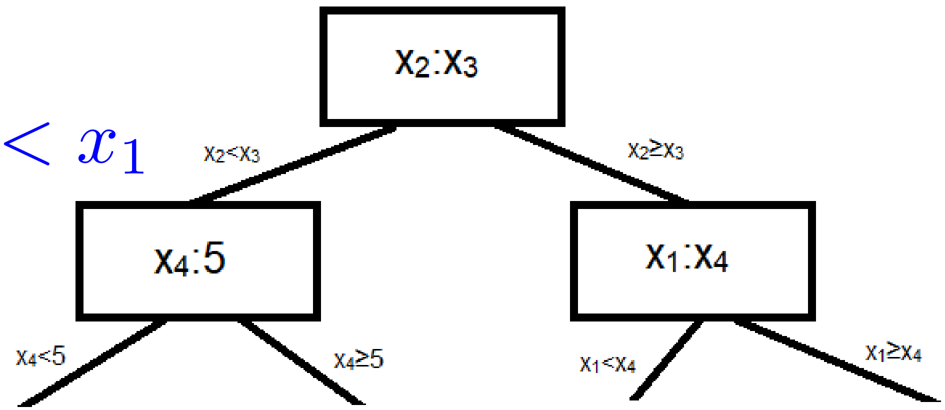


Partitioning n-dimensional Space

Because one unit of work is a comparison, we can use the decision tree model again.

Every leaf will either be YES or NO.

Given an input to this problem, an *ordering* is the actual order of the numbers.



Example: $(7,2,4) \rightarrow x_2 < x_3 < x_1$

$(9,5,1) \rightarrow x_3 < x_2 < x_1$

$(2,3,4) \rightarrow x_1 < x_2 < x_3$

$(3,4,5) \rightarrow x_1 < x_2 < x_3$

YES inputs will not have a unique ordering, but NO leaves will.

What can you say about the set of inputs in any leaf node of the decision tree? Convex

Two NO Inputs in the Same Leaf?

Is it possible to design an algorithm that is so efficient that two NO inputs $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ with *different* orderings wind up in the same leaf?

Assume that $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$ are two inputs with different orderings that wind up in the same NO leaf.

If $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ have the *same* ordering then $\forall i \neq j, x_i > x_j \rightarrow y_i > y_j$.

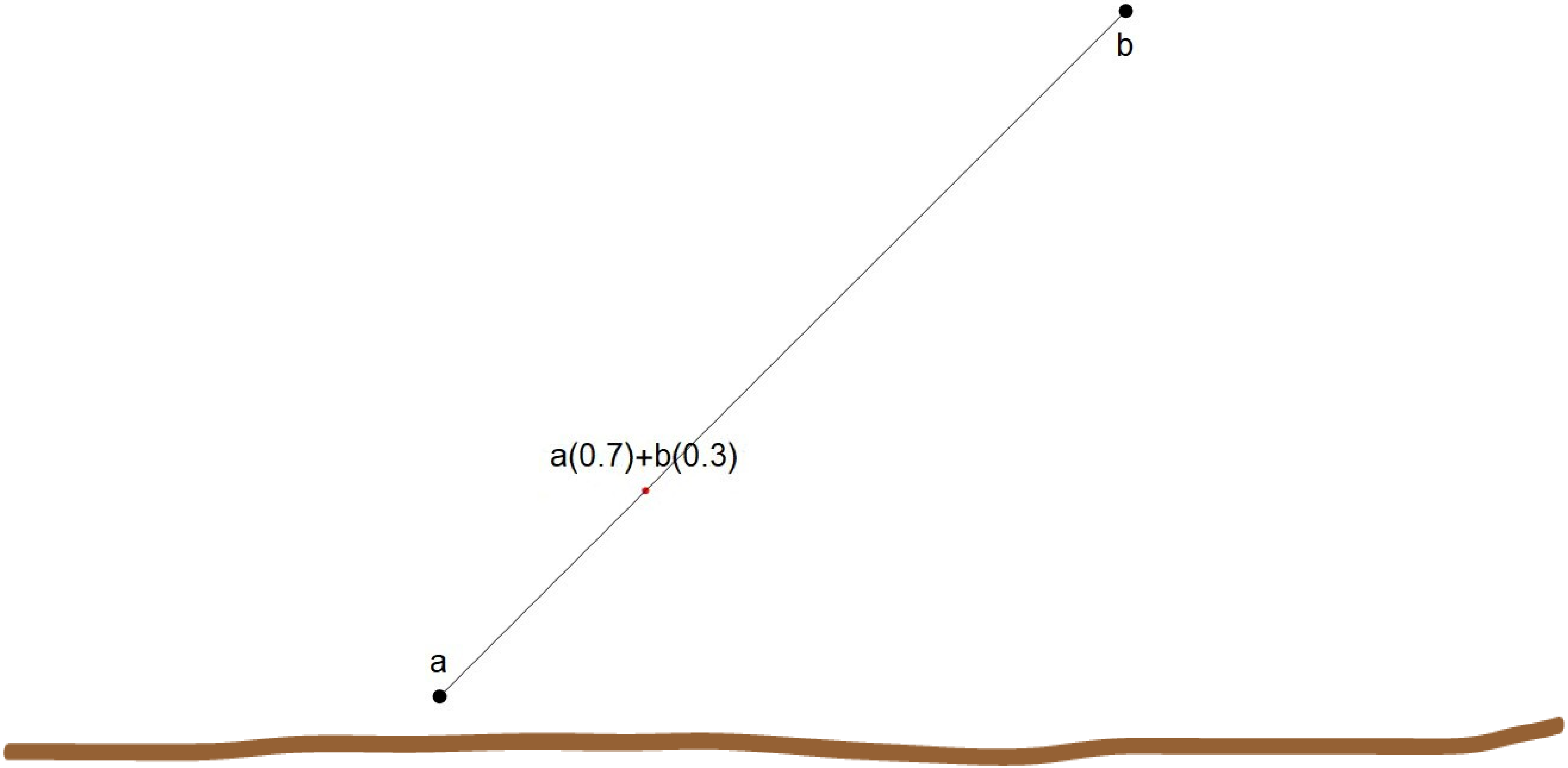
Example: $x = (2, 5, 8)$ and $y = (1, 2, 3)$

Because a and b have *different* orderings,
$\exists i \neq j, a_i > a_j$ and $b_i < b_j$.

Example: $a = (2, 5, 8)$ and $b = (2, 1, 3)$

Two NO Inputs in the Same Leaf?

What does the following expression look like in n-dimensional space if $t \in [0, 1]$? $a(1 - t) + bt$



Two NO Inputs in the Same Leaf?

<u>Math fact</u>: For any $x > 0$, $\exists t \in (0,1)$ such that $f(t) = \frac{1-t}{t} = x$ by letting $t = \frac{1}{1+x}$.

Recall $a_i > a_j$ and $b_i < b_j \Rightarrow \frac{b_i - b_j}{a_j - a_i} > 0$

Math Fact$\rightarrow$ $\exists t \in (0,1)$ such that
$\frac{1-t}{t} = \frac{b_i - b_j}{a_j - a_i} \Rightarrow a_i(1-t) + b_i t = a_j(1-t) + b_j t$

$\Rightarrow \exists$ a point on $\overline{ab}$ such that the ith coordinate and jth coordinate are equal.
$\Rightarrow \exists$ some input on the line from a (NO) to b (NO) where the answer is YES.

But a and b are in the same leaf node and the set of points in any leaf node is CONVEX!

Conclusions

It is impossible for a single NO leaf to contain inputs with distinct orderings.
$\Rightarrow$Every distinct ordering must have its own NO leaf.
$\Rightarrow$There must be at least $n!$ leaves.
$\Rightarrow$The Element Uniqueness Problem is $\Omega(n \log n)$.

Another Surprising Conclusion

Assume that you are given an instance of the Element Uniqueness Problem $(x_1, x_2, \ldots, x_n)$.

Transform it into a Closest Point Problem:
$(x_1, x_2, \ldots, x_n) \to ((x_1, 0), (x_2, 0), \ldots, (x_n, 0))$
Run the fastest possible algorithm that solves the Closest Point Problem on the new problem. Is the answer 0 or not?

YES: At least two of $x_1, x_2, \ldots, x_n$ are the same.
NO: $x_1, x_2, \ldots, x_n$ are all different.
How fast could the Closest Point Problem algorithm have run if the Element Uniqueness Problem is $\Omega(n \log n)$?
The Closest Point Problem is also $\Omega(n \log n)$!