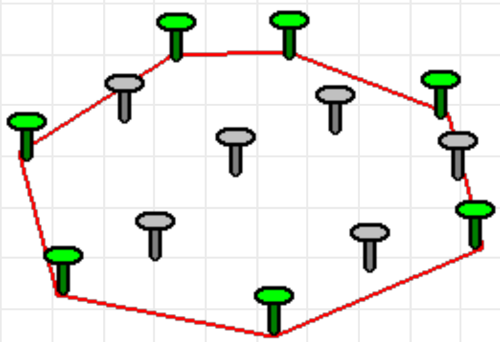
Recall

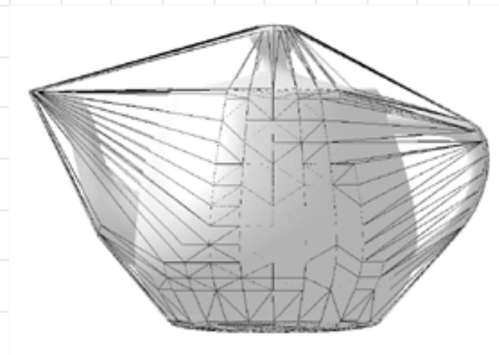
Given points in d-dimensional space, find a good “container” = convex polytope.
Many applications, e.g. collision detection, pattern recognition, motion planning . . .

In 2D, imagine putting a rubber band around the points



https://brilliant.org/wiki/convex-hull/

In 3D, wrap with shrink-wrap



Newton Collision Convex Hull

More formally:

In 2D, the ***convex hull*** of a set of points S is a convex polygon P with vertices in S such that every point of S lies inside.
(definition in 3D and higher later on)

Recall

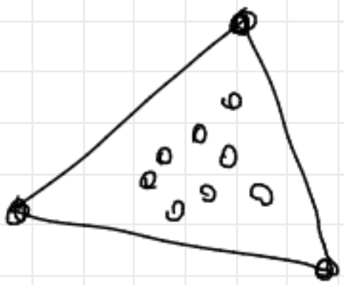
Three O(n log n) time algorithms to find the Convex Hull of n points in 2D

- incremental
- Graham's algorithm
- divide and conquer

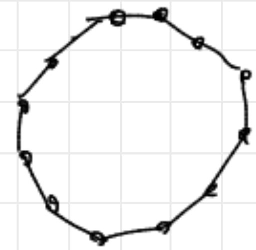
Lower bound of Omega(n log n)

Output sensitive algorithm

Idea:

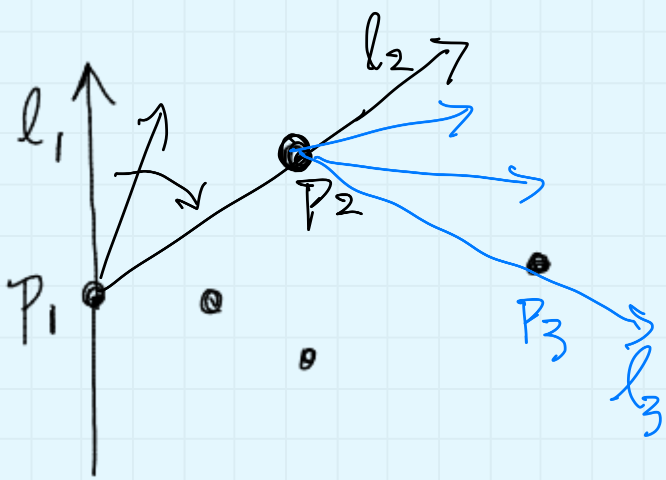


should be faster than



Express the run time as a function of input size, *n*, and output size, *h*.

Gift-Wrapping (or "Jarvis's March")

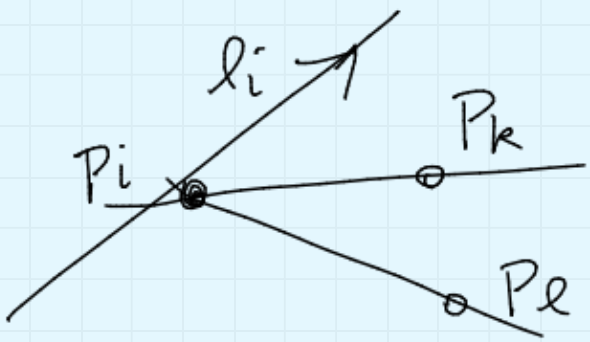


p_1 := min x then max y
l_1 := vertical ray through p_1

"wrap" line l_1 (rotate through p_1) until we hit first point — p_2

Then wrap from p_2

How to "wrap"



to find next point after p_i

like finding min

$O(n)$ to scan through points to get min

To compare

p_k, p_l

use sidedness test

e.g. here p_l is below line $p_i p_k$

so we prefer p_k

Runtime of Gift Wrapping Algorithm:

Time for one wrap step is $O(n)$

Total in worst case $O(n^2)$

As function of n and h — $O(n \cdot h)$

time for one wrap

\# wrap steps

Summary $O(n \log n)$ vs $O(n \cdot h)$

last day

So, what is the best algorithm in terms of *n* and *h*?

O(*n* log *h*) algorithm — first developed by Kirkpatrick and Seidel 1986, uses linear time median finding.

improved by Timothy Chan, 1996.

Chan's Algorithm

Optimal output-sensitive convex hull algorithms in two and three dimensions
TM Chan - Discrete & Computational Geometry, 1996 - Springer
d https://doi.org/10.1007/BF02712873

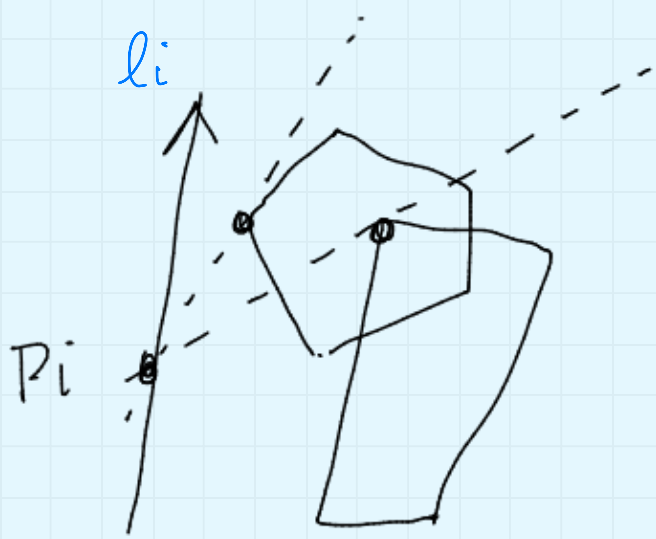
Assume h is known (fix this later)
$m \leftarrow h$

Partition the n points into $\lceil \frac{n}{m} \rceil$ subsets each of size $\leq m$ (arbitrarily)

Find CH of each set e.g. Graham's alg.
so $O(m \log m)$ for each set.

Time so far $O(\frac{n}{m} \cdot m \log m) = O(n \log m)$

Next run Gift Wrapping, but for the wrap step, don't check all n points

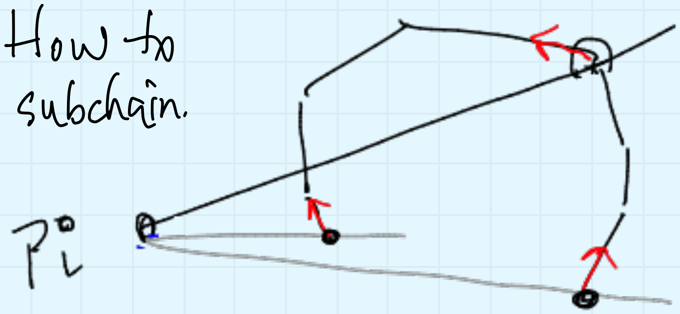


just check the most extreme point of each of the $\frac{n}{m}$ convex sets

extreme = point Pk s.t. angle from li to line PiPk is min.

How to find the most extreme point of a convex hull of size ≤ m.

Ex. How to find subchain.



use binary search idea

in general ~~step~~ have a subchain, test middle & cut chain in half — use sidedness test

Runtime: $O(\log m)$

Time for one wrap step: $O(\frac{n}{m} \cdot \log m)$

polygons

How many wrap steps should we do?

If we do all h wrap steps (h is # ~~pts~~ on CH)

then total time is $O(h \frac{n}{m} \cdot \log m) + O(n \log m)$

great if $m = h$ — we get $O(n \log h)$

How do we find the right *m* ?

Try various values of m. — start small, work up.

Careful: if m is too small

then $O(h \frac{n}{m} \log m)$ can be too big, e.g. $O(hn)$

So stop gift-wrapping after m steps.

Then time to try m is $O(n \log m)$

Note: if we try $m \geq h$ then we find CH.

How do we find the right *m* ?

A related problem:

Search in a sorted but unbounded array of distinct natural numbers.
(in a bounded array A[1 . . k] we can search in log k steps.)

use a doubling technique

try $i = 1, 2, 4, 8, \ldots$

find i s.t. $A(2^i) < x < A(2^{i+1})$

goal

$\log x$

Then use binary search in bounded range $\log x$

How do we find the right *m* ?

Try an increasing sequence of values of m until we get one bigger than h (i.e., one where the algorithm find the CH)

try doubling $m = 1, 2, 4, 8, \ldots \quad m = 2^i$

Time $\sum_{i=1}^{\log h} n \log 2^i = n \sum_{i=1}^{\log h} i = O(n \log^2 h)$

Too big!

want $O(n \log h)$

$2^i \geq h \quad i \geq \log h$

Try $m = 2, 4, 16 \quad m = 2^{2^i}$

Time $\sum_{i=1}^{\log\log h} n \log(2^{2^i}) = n \sum_{i=1}^{\log\log h} 2^i = O(n \overbrace{\log h}^{2^{\log\log h}})$

$2^{2^i} \geq h \quad i \geq \log\log h$

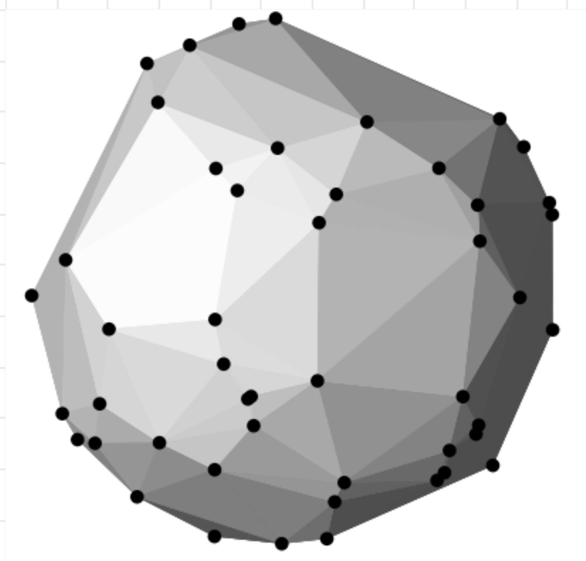
YES!

What's next?

- definitions of convex hull in any dimension, and more about convex polyhedra

- divide and conquer for convex hull in 3D

- randomized algorithm for convex hull in any dimension



Equivalent definitions of ***Convex Hull*** of a set of points S

1. intersection of all half-spaces containing S

2. all *convex combinations* of points in S

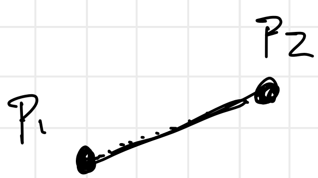
3. A *convex polyhedron* P (polygon in 2D) with vertices from S and such that all points of S are inside P

A set C is ***convex*** if for all points p, q in C, the line segment pq is in C.

not convex

A ***convex combination*** of points p1, p2, . . , pn is

$$\sum_{i=1}^{n} \lambda_i p_i \quad \text{for} \quad \sum_{i=1}^{n} \lambda_i = 1, \;\; \lambda_i \geq 0$$



$\lambda_1 p_1 + \lambda_2 p_2 \quad 0 \leq \lambda_i \leq 1$

$\lambda_1 + \lambda_2 = 1$

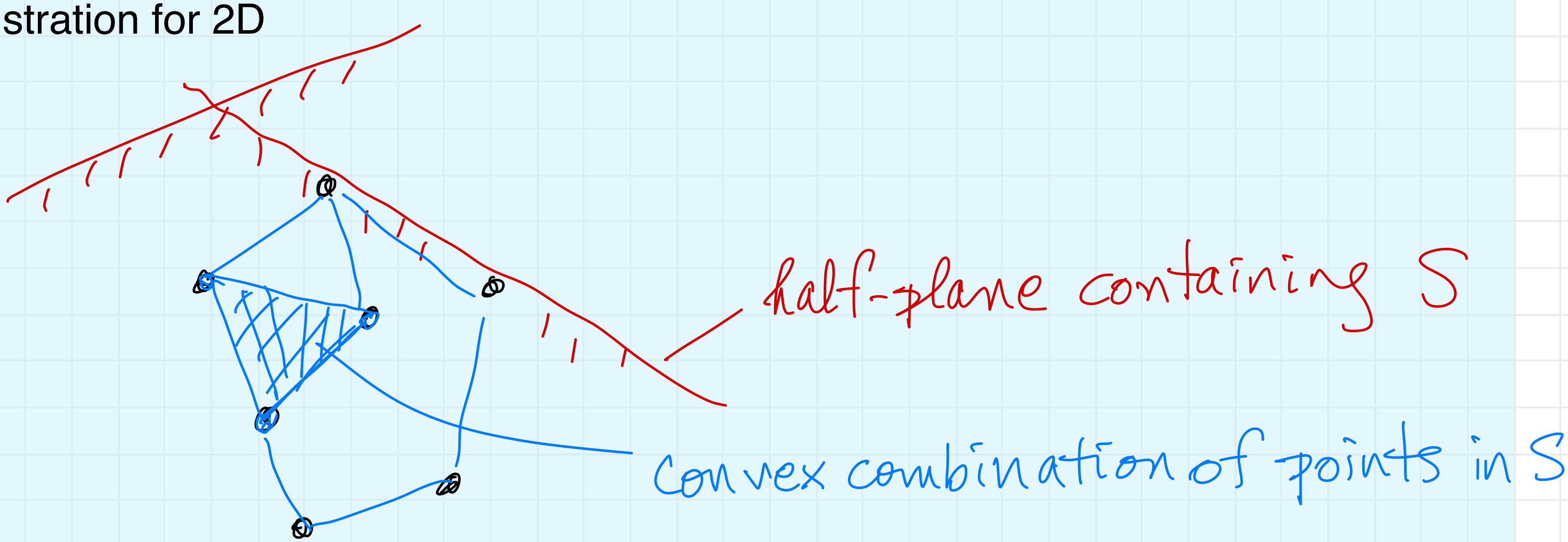
Equivalent definitions of ***Convex Hull*** of a set of points S

1. intersection of all half-spaces containing S

2. all *convex combinations* of points in S

3. A *convex polyhedron* P (polygon in 2D) with vertices from S and such that all points of S are inside P

Illustration for 2D



Equivalent definitions of ***Convex Hull*** of a set of points S — CH(S)

1. intersection of all half-spaces containing S

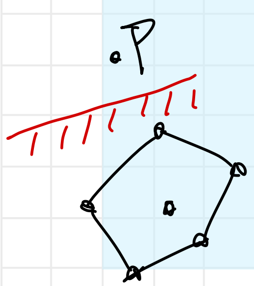
2. all *convex combinations* of points in S

3. A *convex polyhedron* P (polygon in 2D) with vertices from S and such that all points of S are inside P

How are these related?

1 and 2 are positive and negative (NP and co-NP) characterizations.

- if p is in CH(S), show how p is a convex combination of points in S
- if p is outside CH(S), show a half-~~pla~~space separating p from CH(S)



Equivalence of 1 and 2 is proved using some version of Farkas's Lemma

either p is a convex combination of points of S
OR
there is a plane separating p from S
AND NOT BOTH

This is a fundamental result for Linear Programming. W https://en.wikipedia.org/wiki/Farkas%27_lemma

Equivalent definitions of ***Convex Hull*** of a set of points S

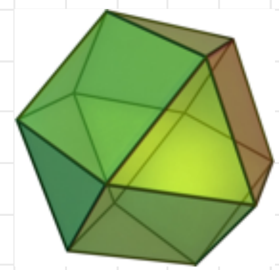
1. intersection of all half-spaces containing S

2. all *convex combinations* of points in S

3. A *convex polyhedron* P (polygon in 2D) with vertices from S and such that all points of S are inside P

A ***convex polyhedron*** (in dimension *d*) is a bounded intersection of half-spaces

$$\{x \in \mathbb{R}^d : Ax \leq b\}, \quad A = m \times d \text{ matrix}, b = m \text{ vector}$$

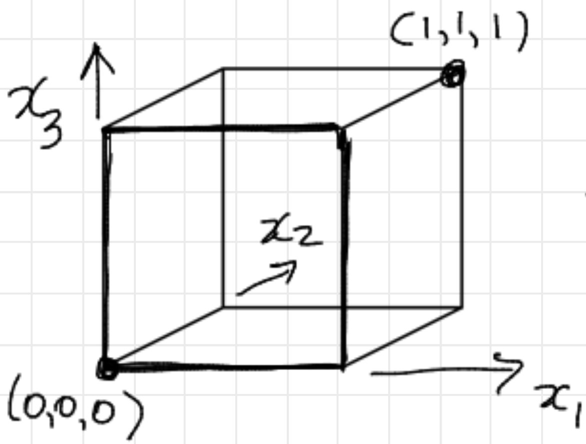


Caution: the term “polyhedron” means different things in different areas (convex/non-convex, bounded/unbounded)

A polyhedron has $\underbrace{\text{vertices, edges}}_{2D}$, faces, . . .

$\underbrace{\text{vertices, edges, faces}}_{3D}$

Face of a polyhedron



A cube in 3D is the intersection of 6 half-spaces

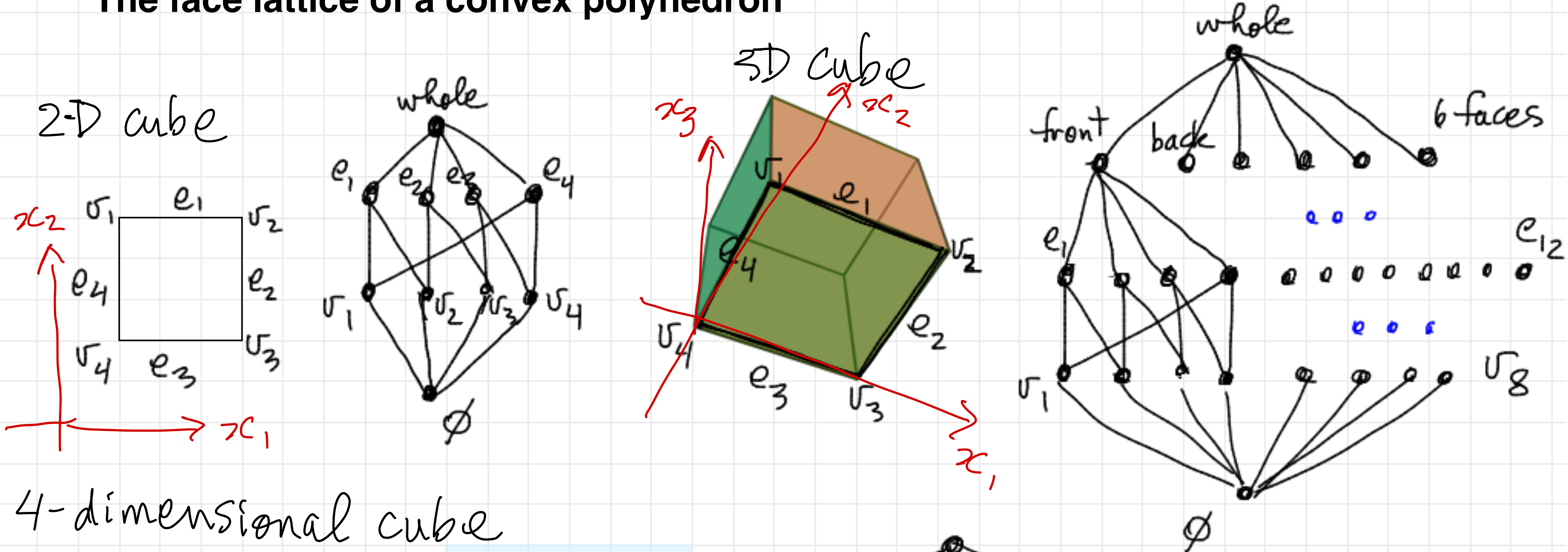
$$\{(x_1, x_2, x_3) : 0 \le x_1 \le 1,\ 0 \le x_2 \le 1,\ 0 \le x_3 \le 1\}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \le \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

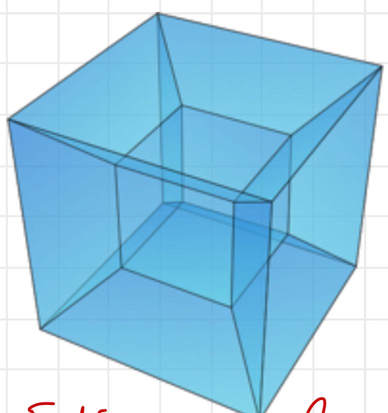
A *face* is $\{x \in \mathbb{R}^d : Ax \le b\}$ and some inequalities are changed to equalities.

e.g. front face of cube has $x_2 = \cancel{1}.$ 0

The face lattice of a convex polyhedron



4-dimensional cube

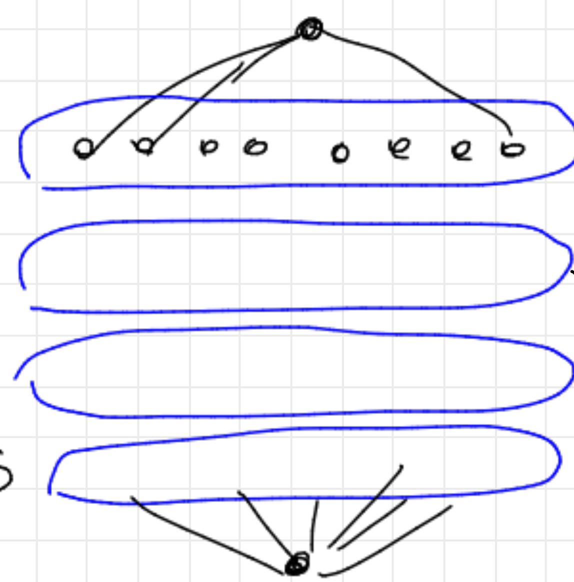


$\underline{8}$ cubes

$\underline{24}$ squares

$\underline{32}$ edges

$\underline{16}$ vertices



$x_1 \quad x_2 \quad x_3 \quad x_4$

$\binom{4}{2} \cdot 4 = 6 \cdot 4$

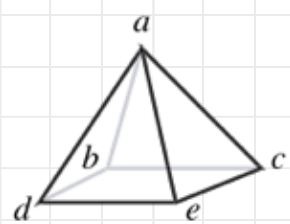
$\binom{4}{3} \cdot \boxed{8} =$

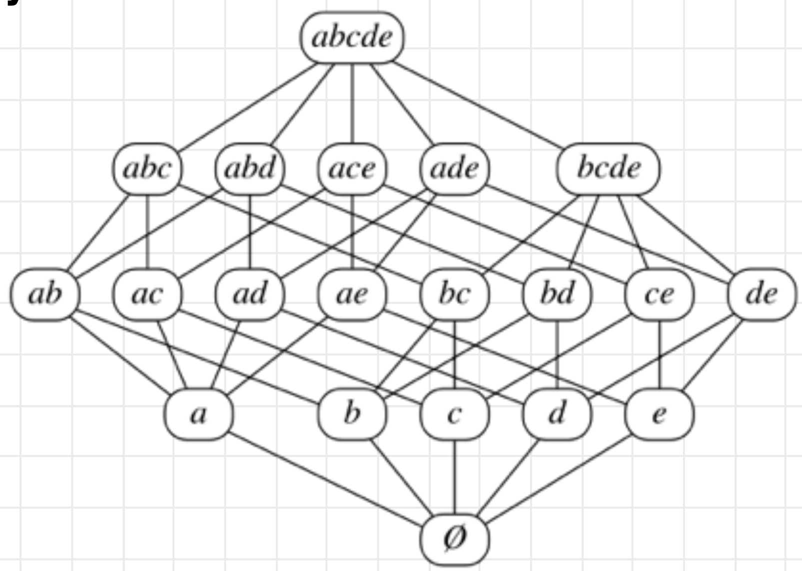
2^3

EX. 5 dimensional cube

EX. — work this out.

The face lattice of a convex polyhedron





i-face = i dimensional face

0-face = vertex

1-face = edge

⋮

(d-1)-face = <u>facet</u>

d-face = the whole polyhedron.

Equivalent definitions of ***Convex Hull*** of a set of points S

1. intersection of all half-spaces containing S

2. all *convex combinations* of points in S

3. A *convex polyhedron* P (polygon in 2D) with vertices from S and such that all points of S are inside P

Equivalence of these definitions proved using:

Theorem [Minkowski, Weyl] The set of all convex combinations of p1, . . . , pn is a bounded convex polyhedron whose vertices are a subset of p1, . . . , pn

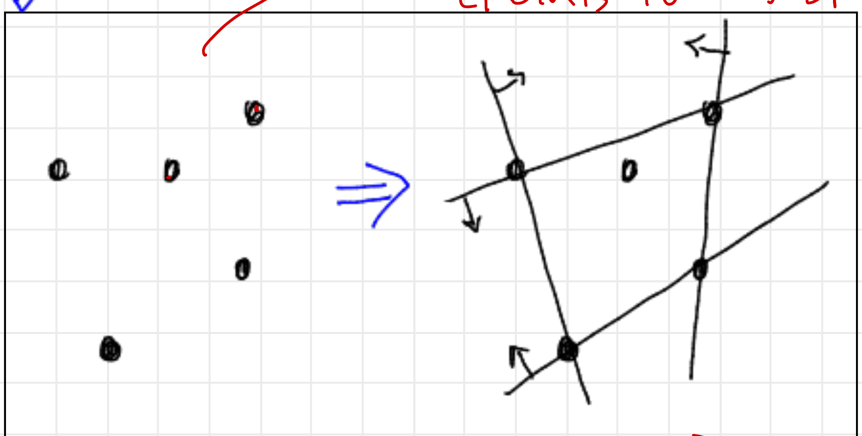
The Convex Hull problem and its Dual

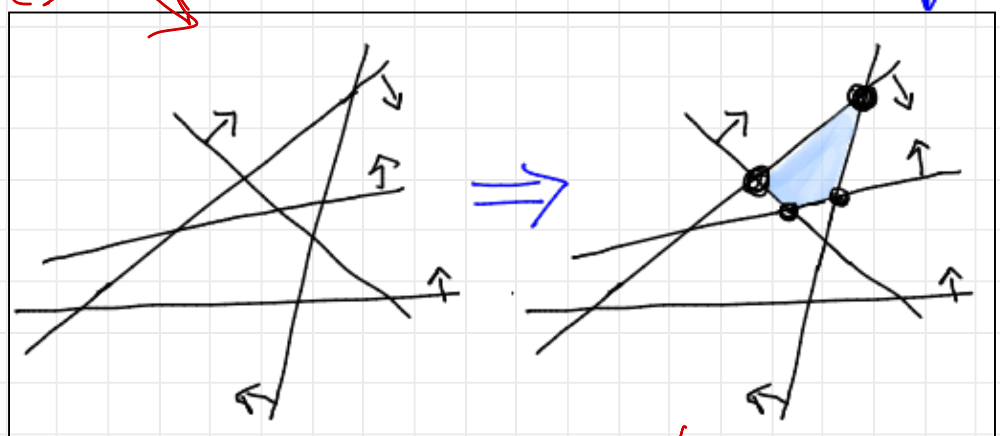
The Convex Hull problem. Given a set of *n* points in *d*-dimensions, find their convex hull (as an intersection of half-spaces). Or sometimes we ask for the whole face lattice.

Dual problem. Given a set of *m* halfspaces in *d*-dimensions, find their intersection (as the set of vertices of the polyhedron). Or sometimes we ask for the whole face lattice.

In fact, these two problems are the same by a ***duality*** map. — maps points ⟷ half-spaces

duality map.
(points to half-spaces.





duality map
(points → half-spaces)