FCT/Unesp – Presidente Prudente
Departamento de Matemática e Computação

Projeto e Análise de Algoritmos: Divisão e Conquista

Prof. Danilo Medeiros Eler
danilo.eler@unesp.br

(apresentação adaptada: ver referências no final)

Departamento de Matemática e Computação
Faculdade de Ciências e Tecnologia
Unesp – Universidade Estadual Paulista
Presidente Prudente/SP, Brasil

DEPARTAMENTO DE MATEMÁTICA E COMPUTAÇÃO

Divisão e Conquista

Procedimento básico

1. Dividir o problema em partes menores
2. Resolver o problema para essas partes (supostamente mais fácil, ou até trivial)
3. Combinar em uma solução global

- Geralmente leva a soluções eficientes e elegantes, muitas vezes de natureza recursiva
- Está normalmente relacionado a uma equação de recorrência que contém termos referentes ao próprio problema

Divisão e Conquista

Recorrência

$$T(n) = aT\left(\frac{n}{b}\right) + f(n),$$

onde:

- a indica o número de sub-problemas gerados
- b o tamanho de cada um dos problemas
- $f(n)$ o custo de resolver cada sub-problema

Exemplos

- Ordenação: mergesort, quicksort
- Busca: busca binária

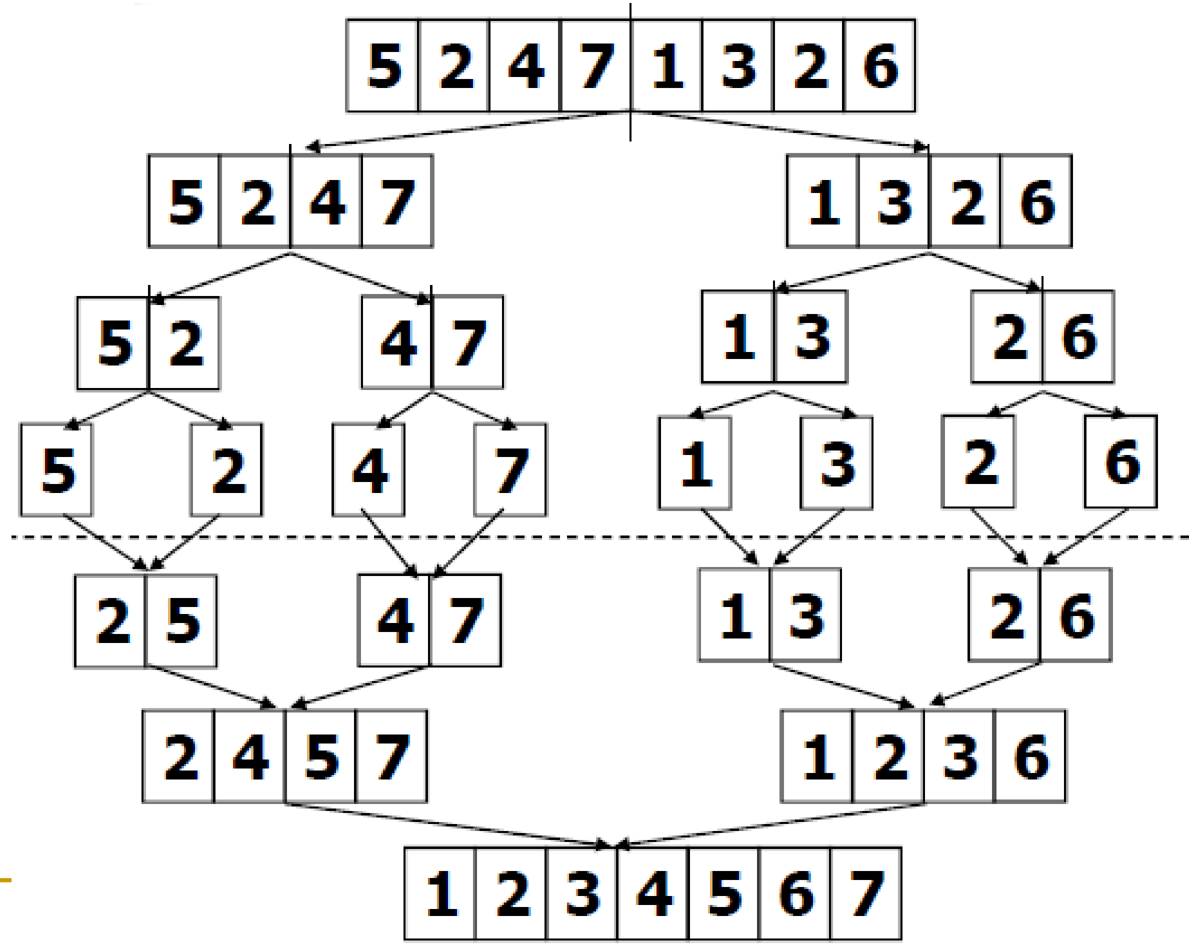
Divisão e Conquista

- Não é aplicado unicamente a problemas recursivos
- Em geral as estratégias de resolução de problemas envolvem um dos cenários abaixo:
  - Processar separadamente partes do conjunto, mas a solução de uma parte influencia no resultado de outra
  - Eliminar partes dos dados a serem processados

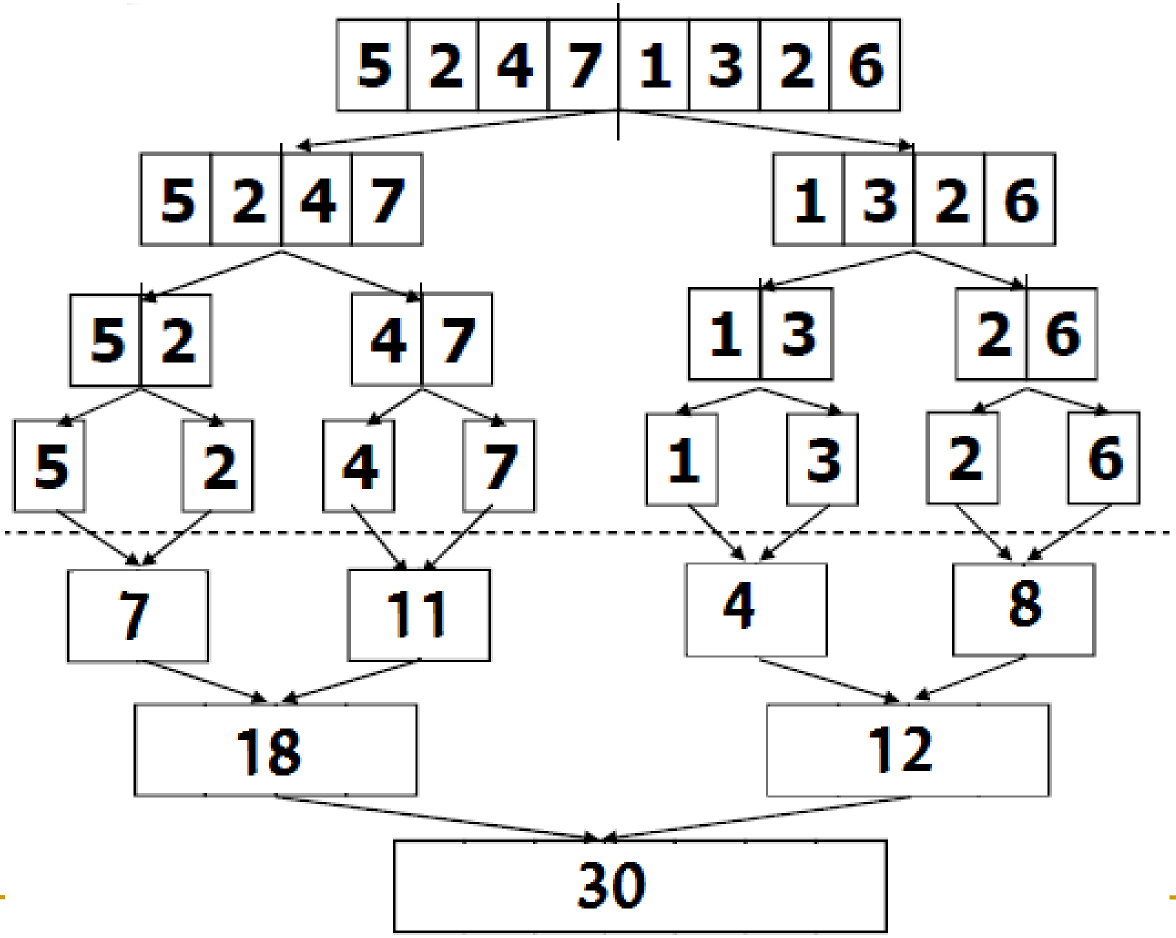
Exemplo 1: Merge-Sort

Processar separadamente partes do conjunto

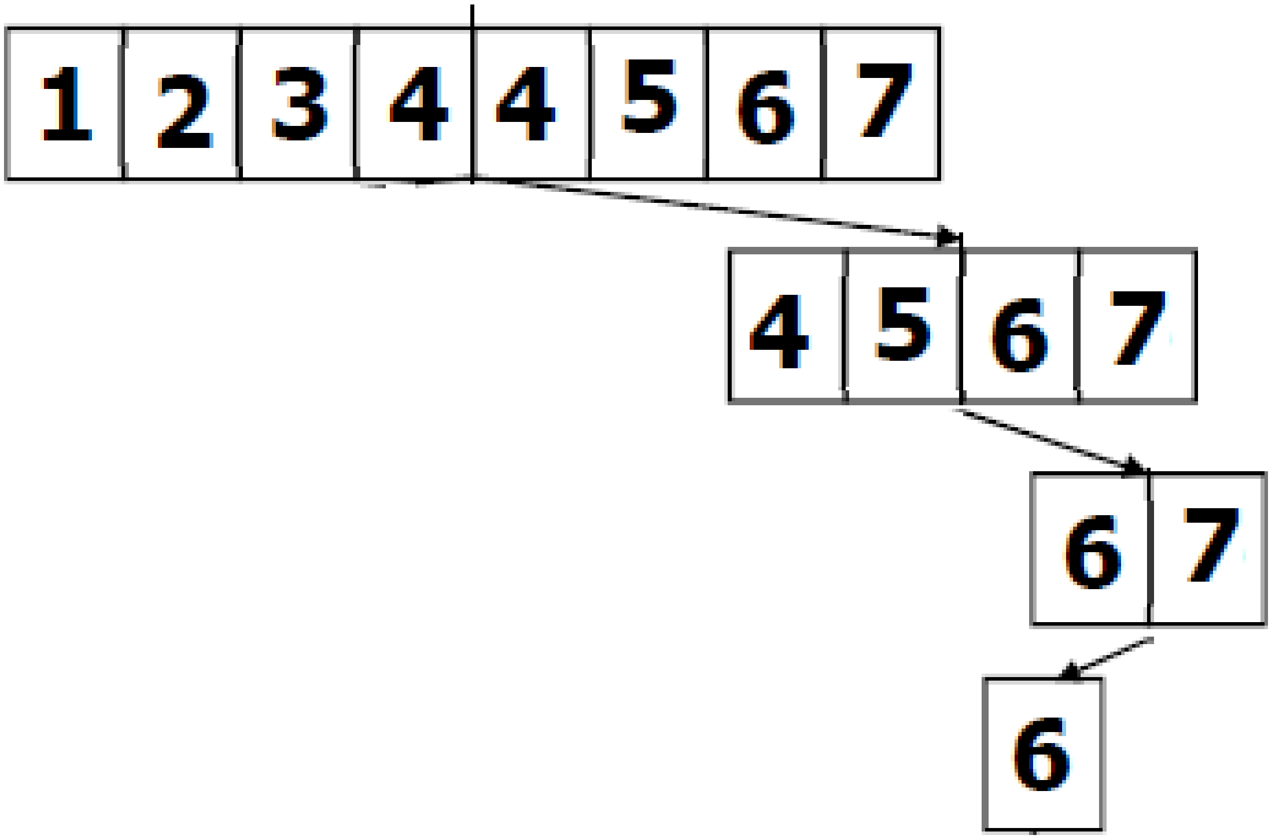
A solução de uma influência na de outra



Exemplo 2: soma
Processar separadamente partes do conjunto



Exemplo 3: busca binária
Eliminar partes do conjunto de dados

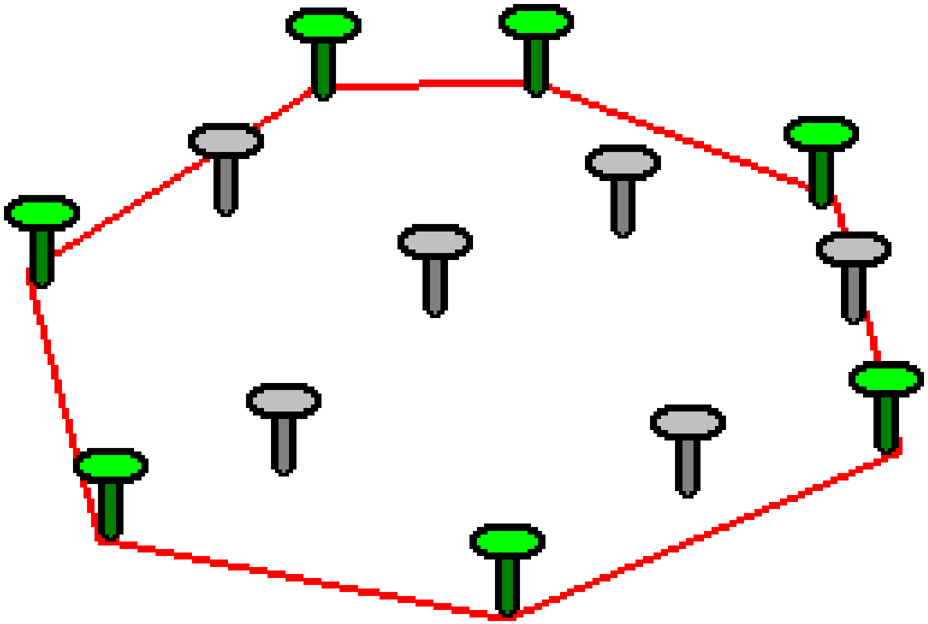


Convex Hull

- Convex Hull (Fecho Convexo)
- É um dos problemas mais interessantes da geometria computacional
- O objetivo é encontrar o fecho convexo que englobe um conjunto de pontos no plano cartesiano

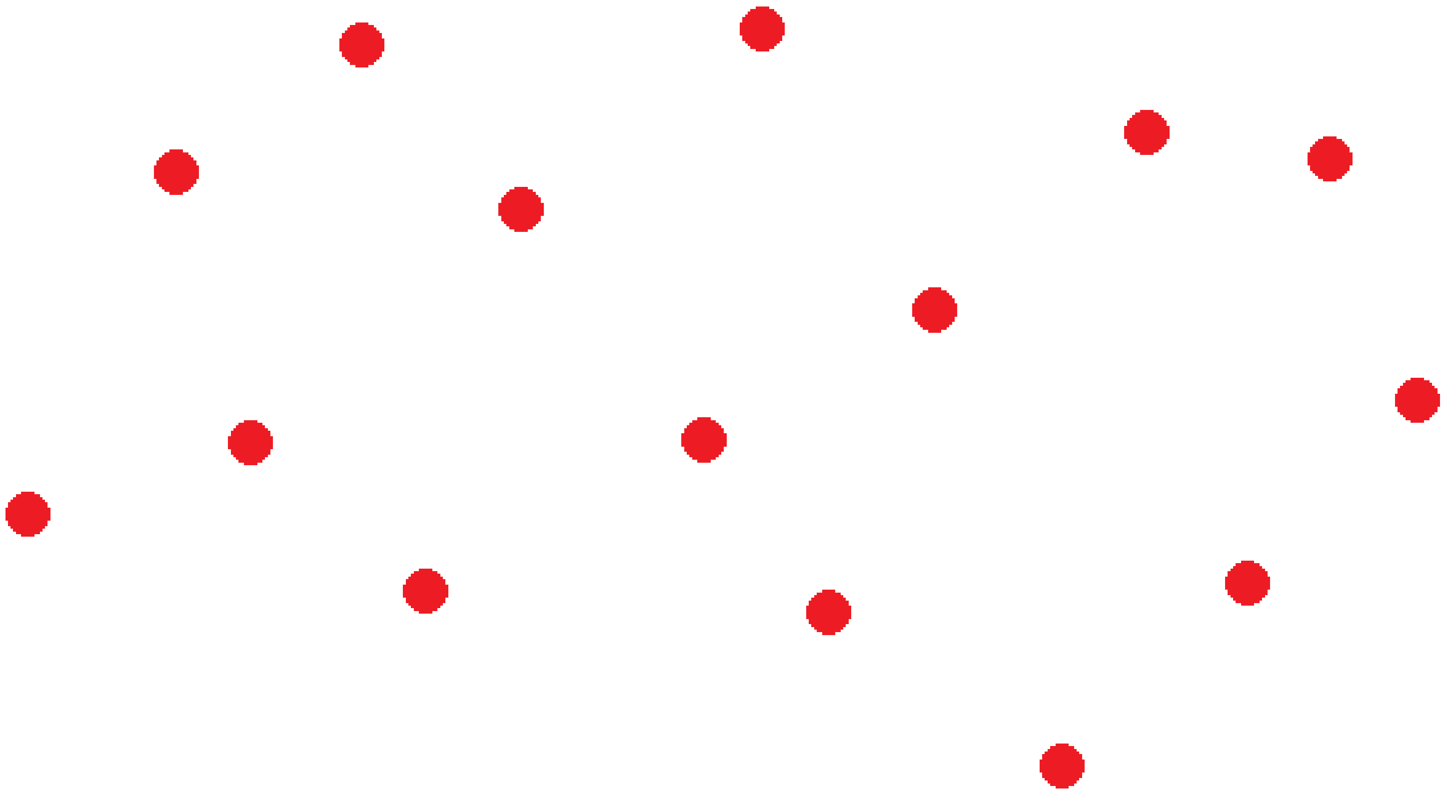
Convex Hull

- Se tivéssemos um conjunto de pregos e soltássemos um elástico em volta, os pregos envolvidos pelo elástico formam o fecho convexo desse conjunto



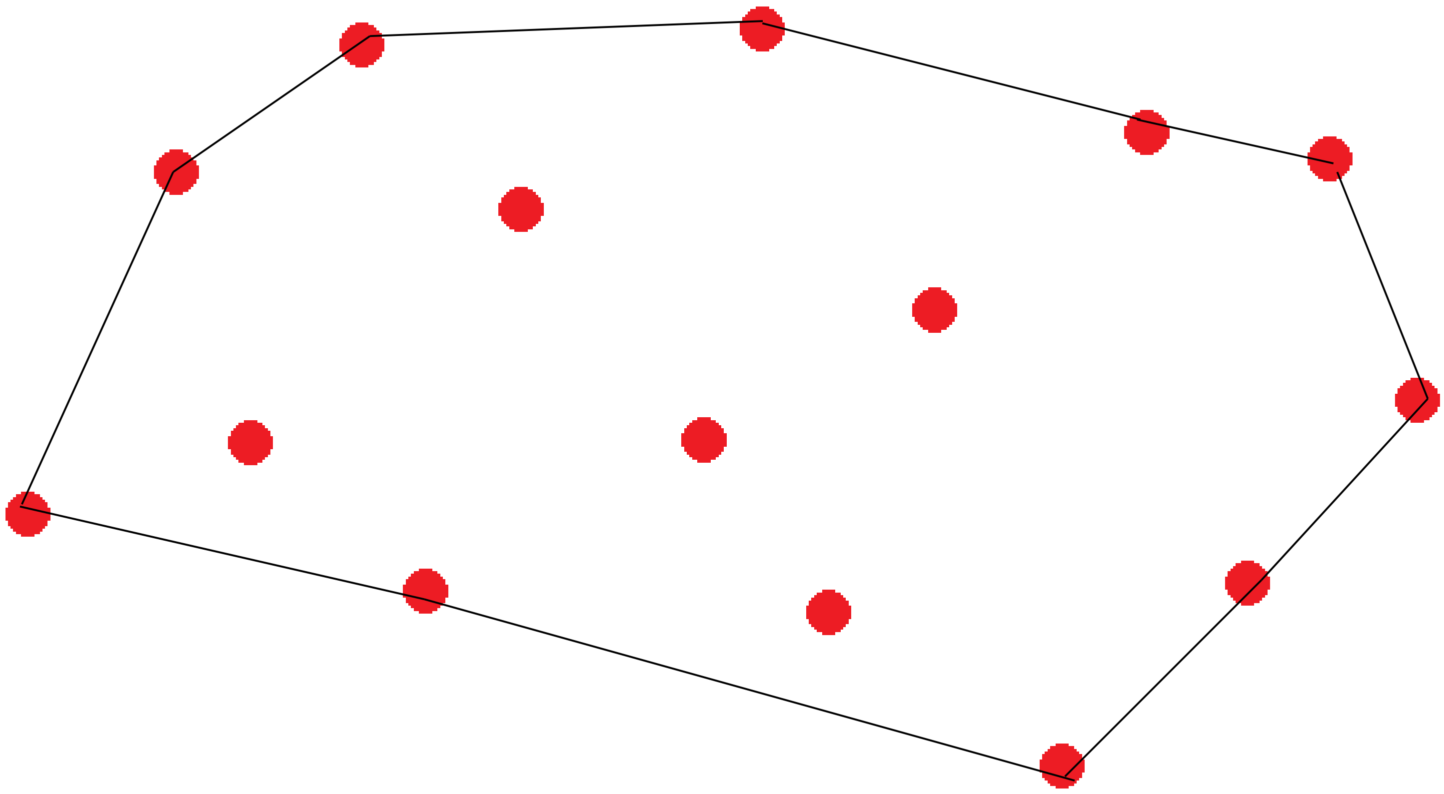
Convex Hull

- Dado um conjunto de pontos



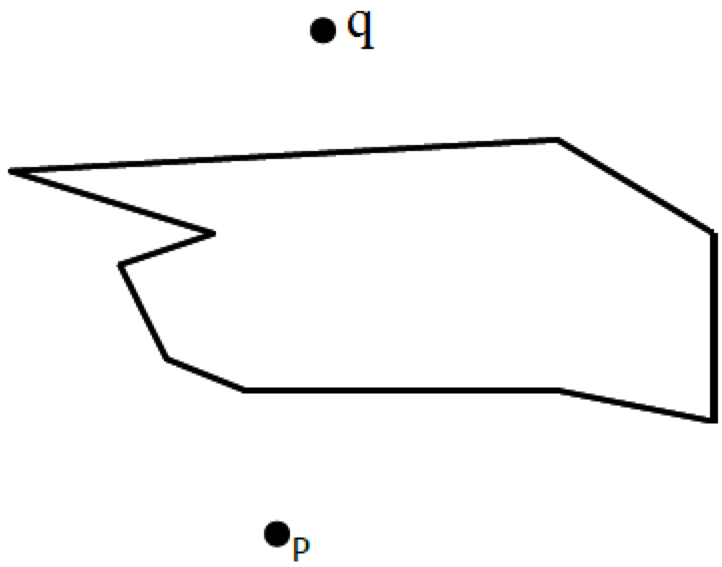
Convex Hull

- O fecho convexo é o menor polígono convexo que contém todos os pontos do conjunto



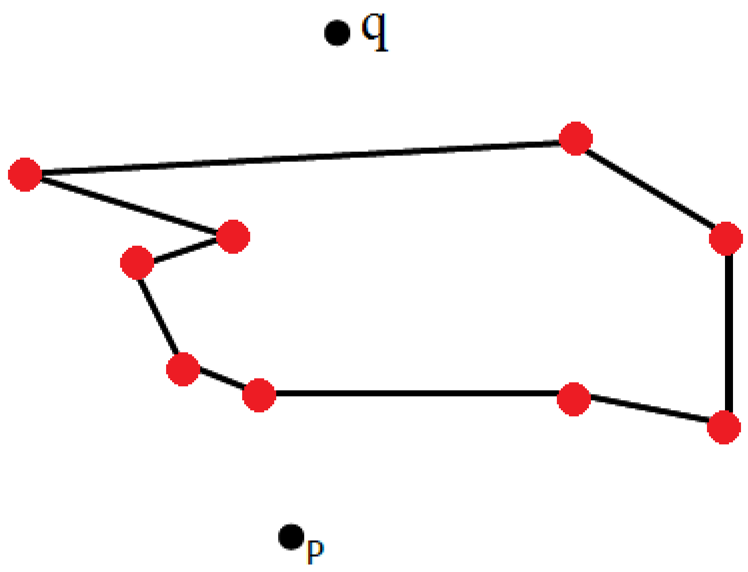
Convex Hull

- Exemplo: minimizar o caminho de **p** a **q**, sem pular o obstáculo



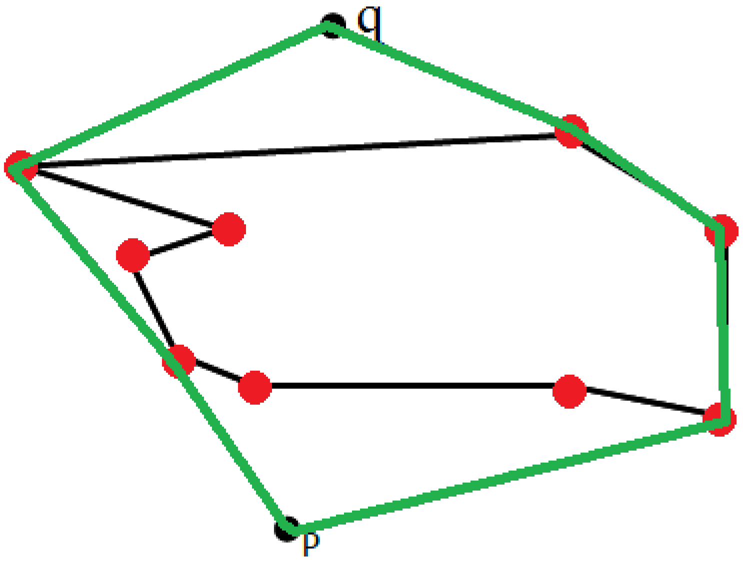
Convex Hull

- Exemplo: minimizar o caminho de **p** a **q**, sem pular o obstáculo



Convex Hull

- Exemplo: minimizar o caminho de **p** a **q**, sem pular o obstáculo



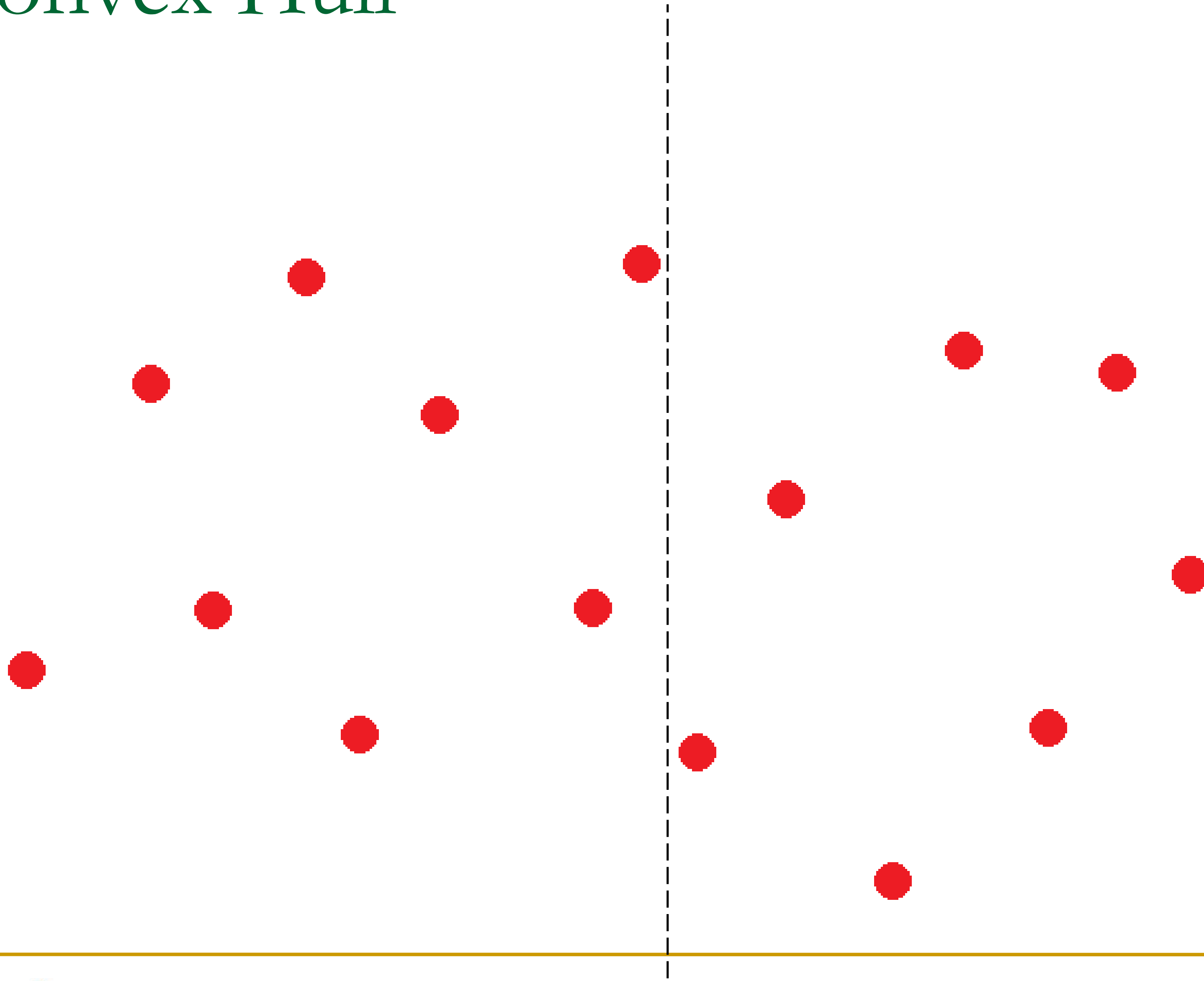
Convex Hull

- Uma maneira rápida de calcular o fecho convexo é por meio da técnica de divisão e conquista
- Dividimos o conjunto de pontos ao meio até chegar em um conjunto com 3 pontos ou menos
  - Então criamos de maneira simples o fecho desse subconjunto
- Em seguida, a etapa de conquista, consiste em unir os fechos previamente criados

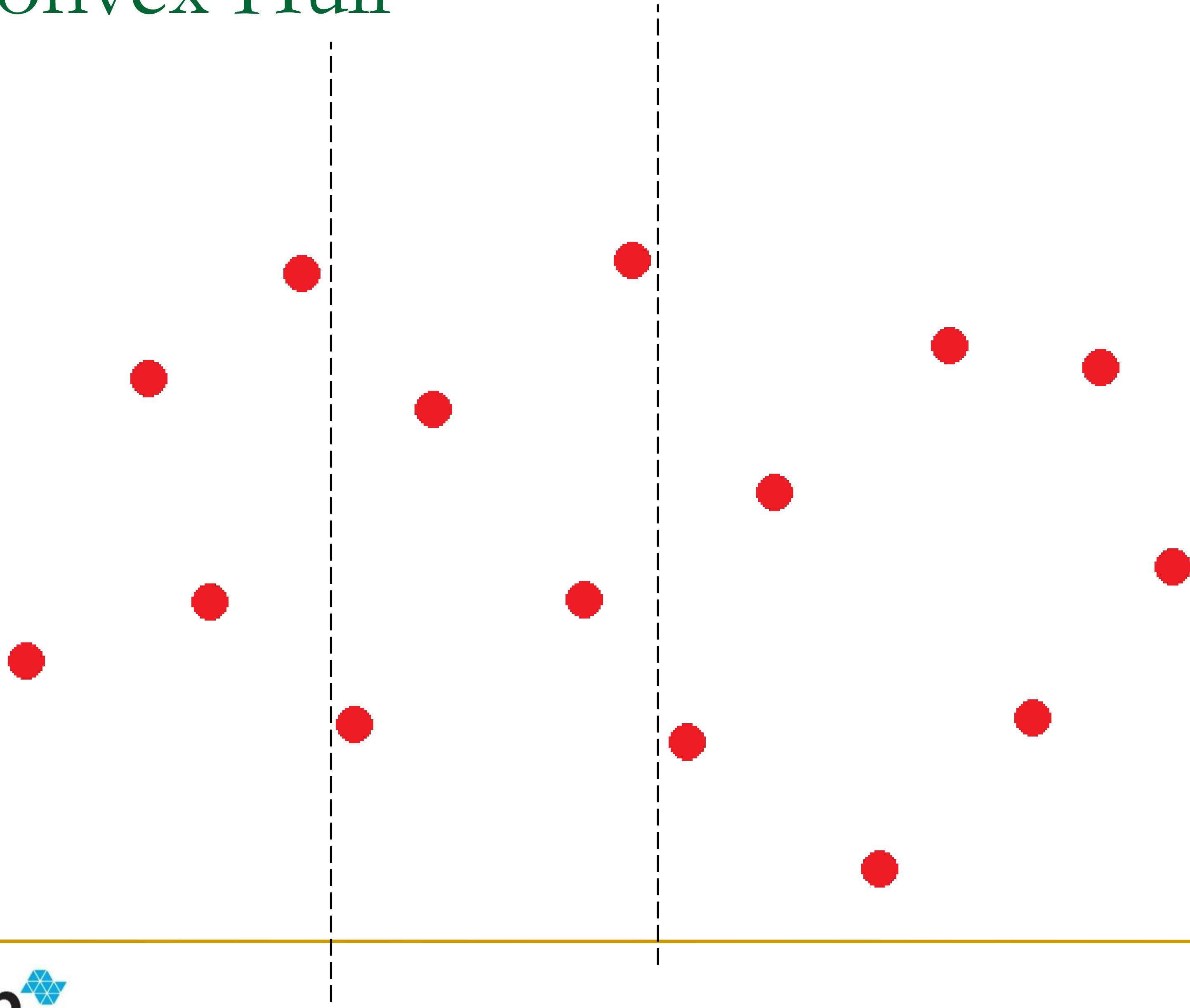
Convex Hull

```
convexHull(pontos, ini, fim)
  se (fim - inicio + 1) > 3) {
      int meio = (inicio + fim) / 2;
      Poligono esq = convexHull(pontos, ini, meio);
      Poligono dir = convexHull(pontos, meio + 1, fim);
      retorna merge(esq, dir);
  else
      retorna criaPoligonoBase(pontos, ini, fim);
   fimse
fim
```

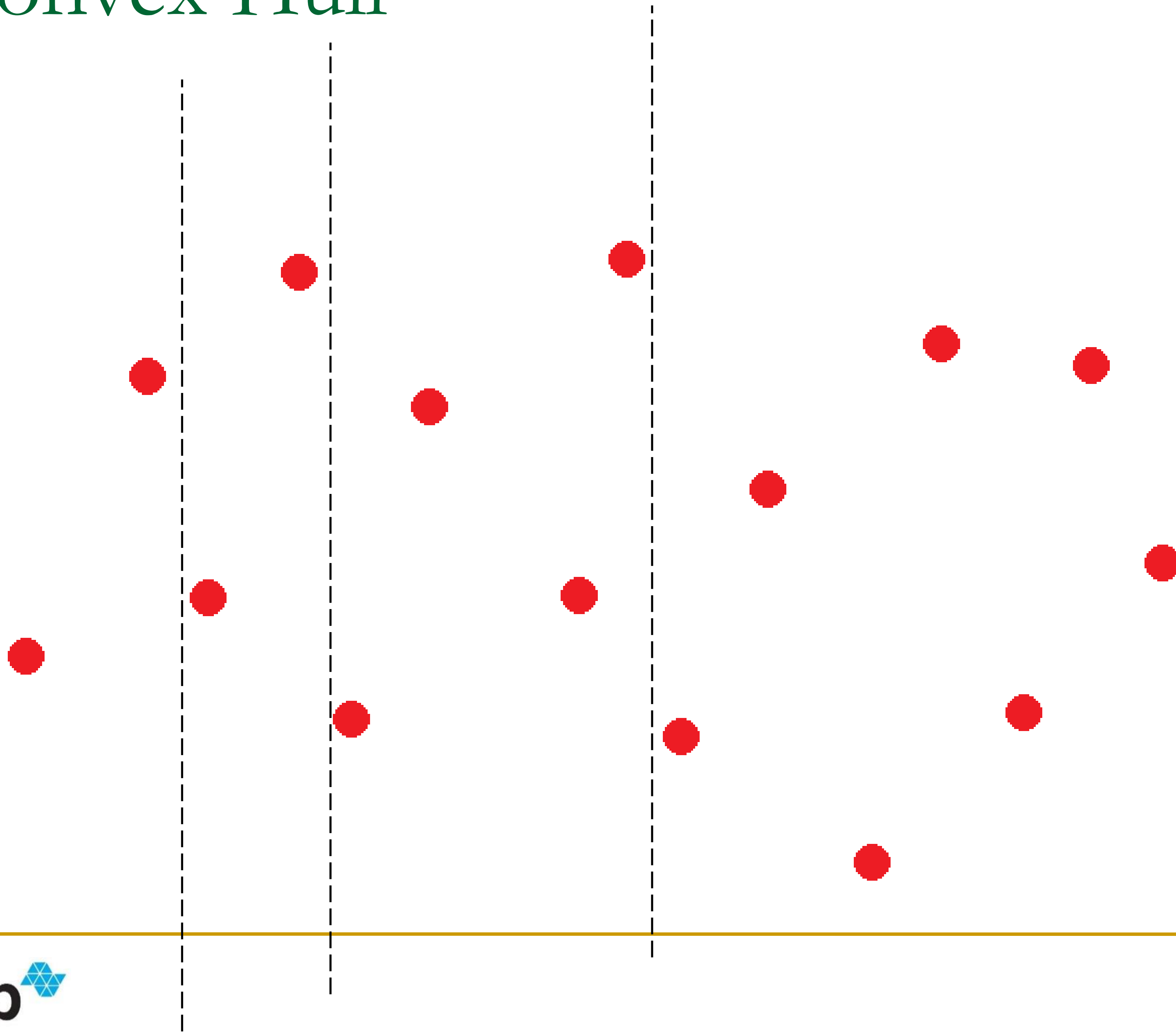
Convex Hull



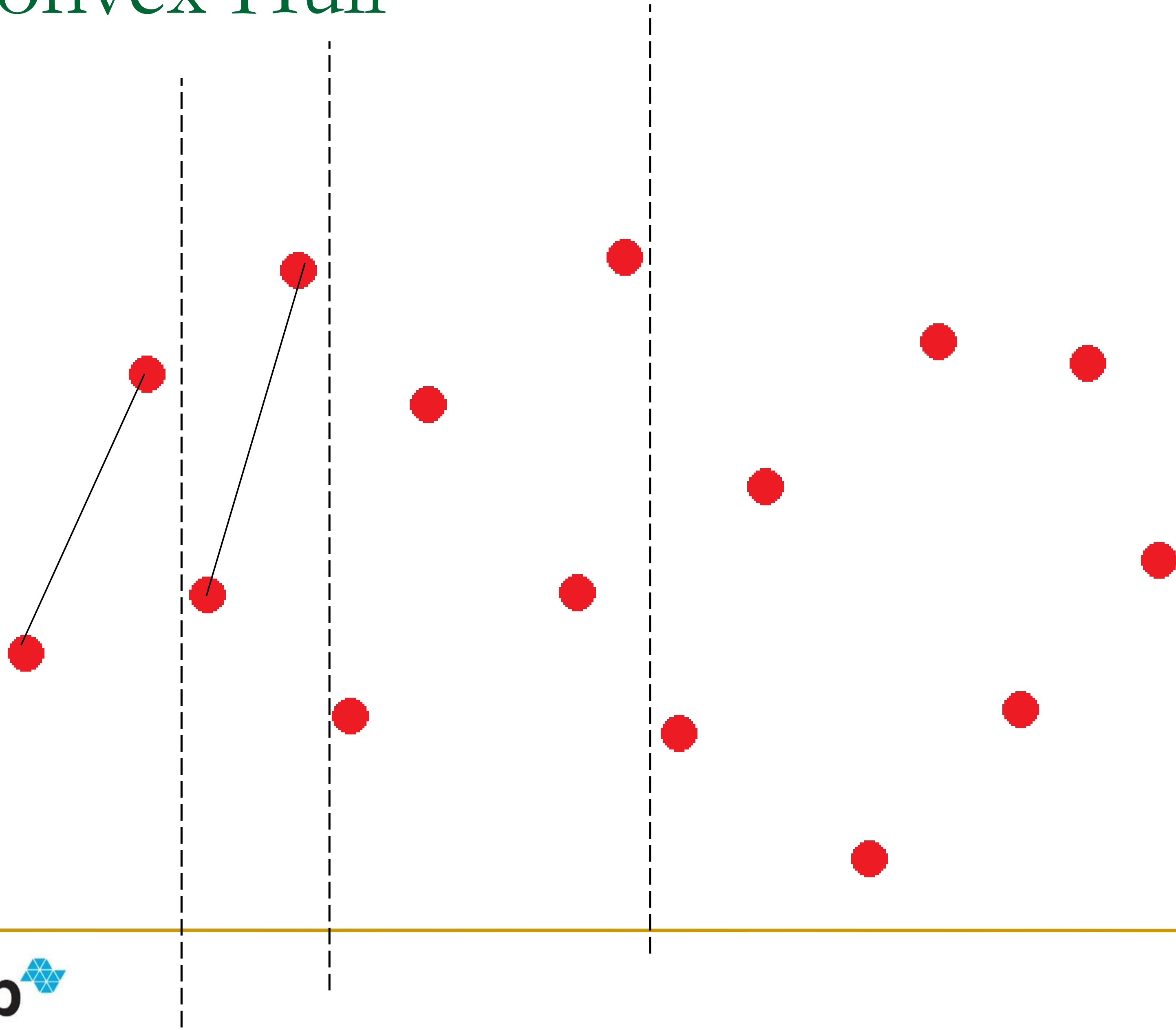
Convex Hull



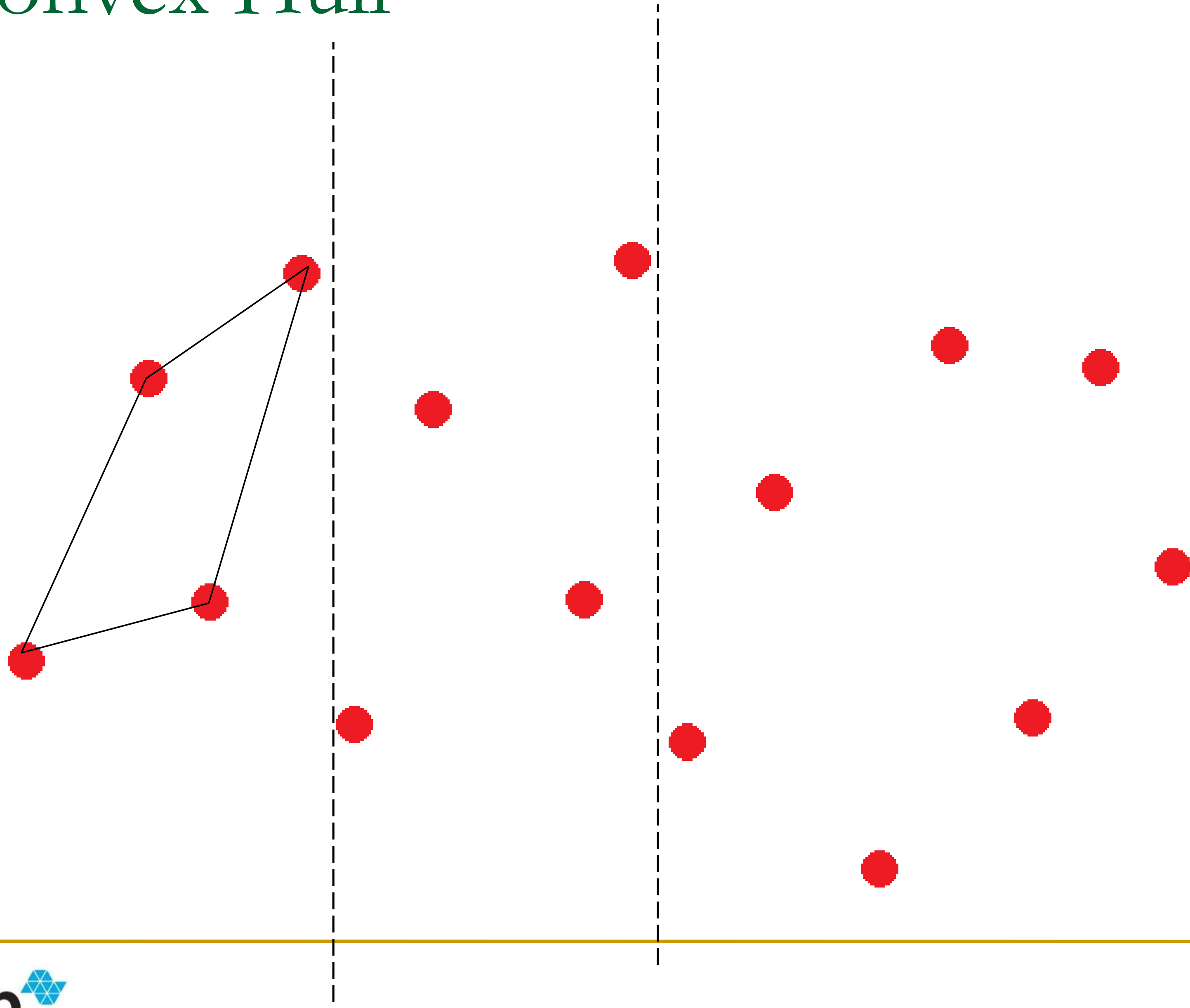
Convex Hull



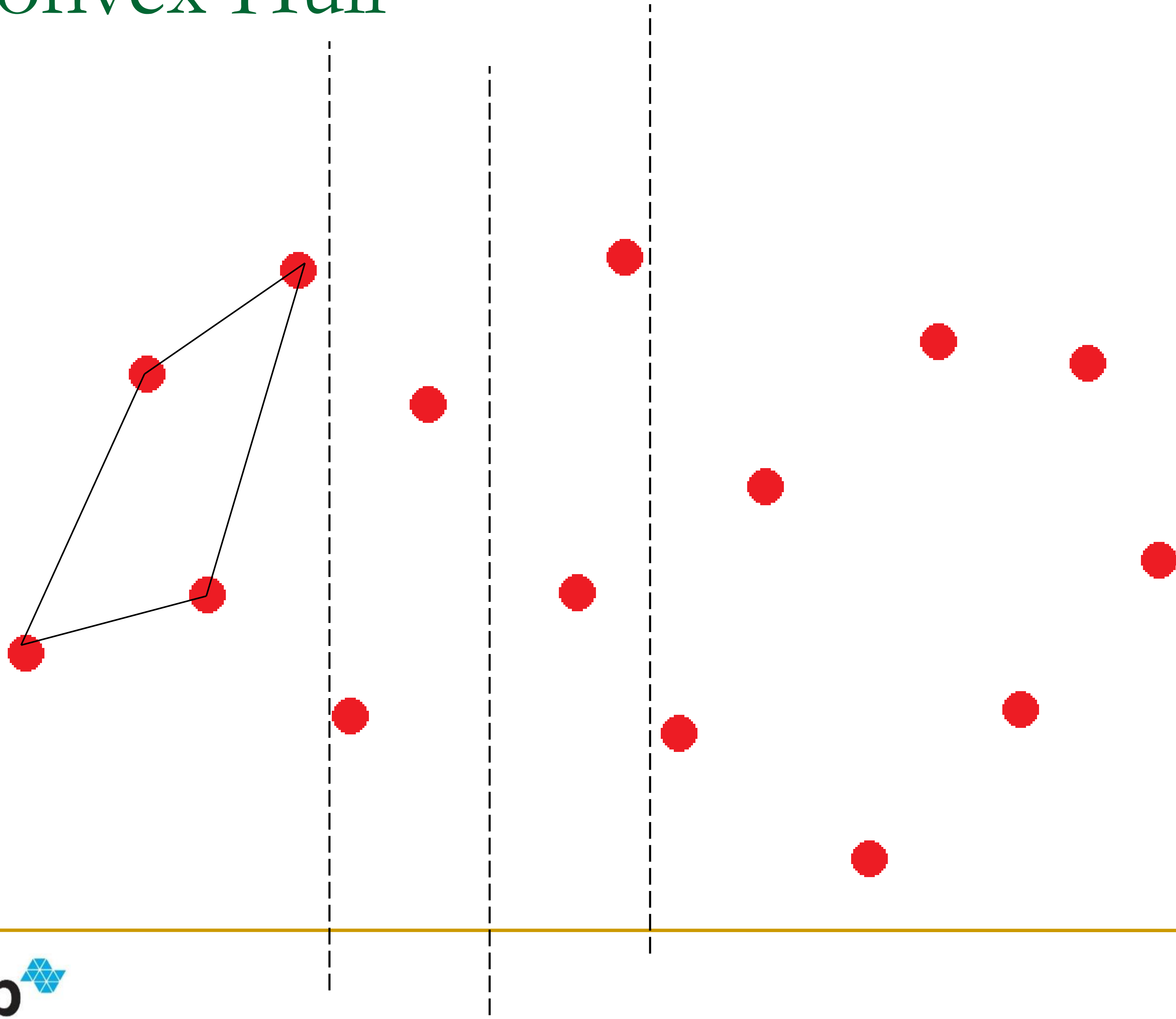
Convex Hull



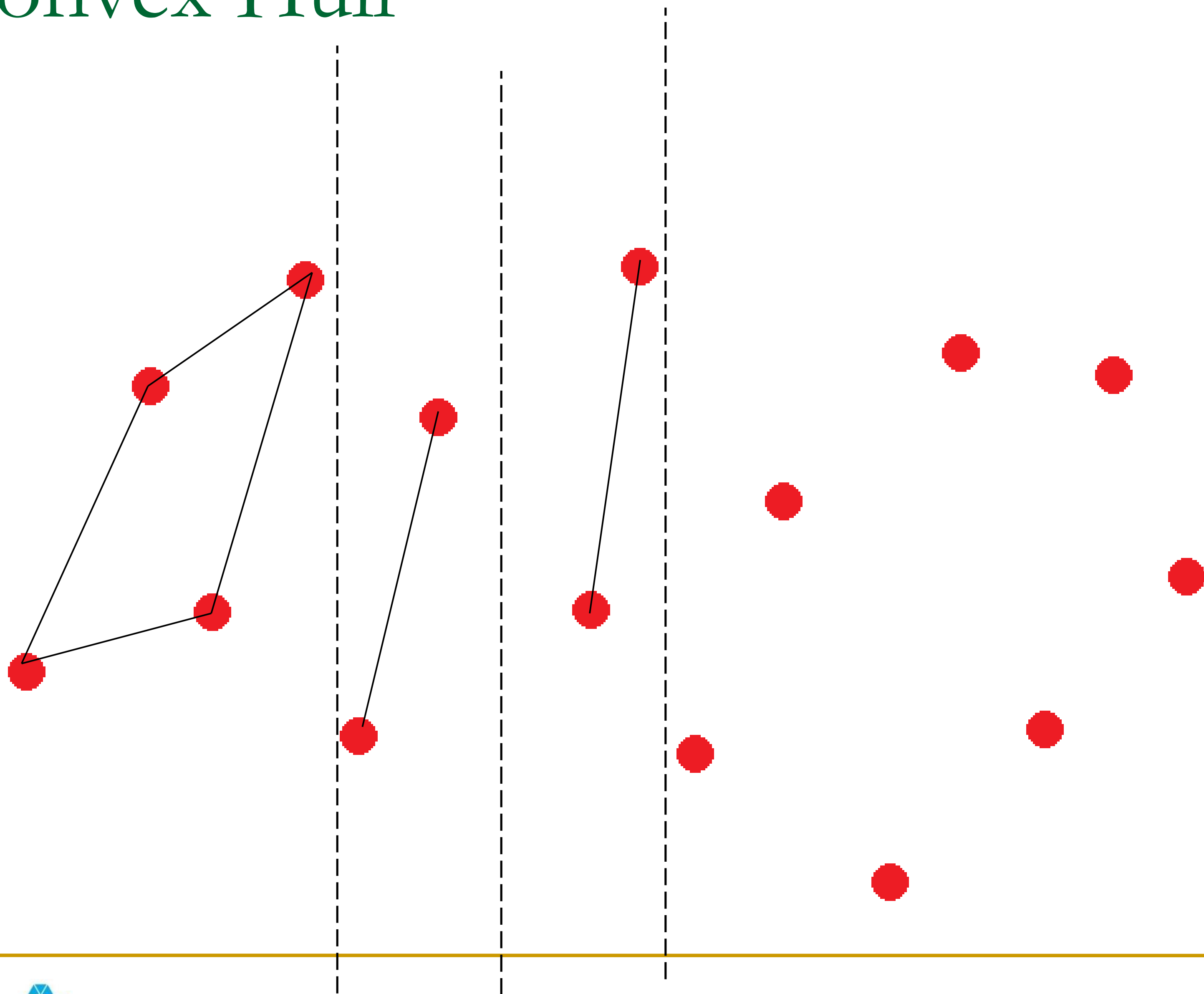
Convex Hull



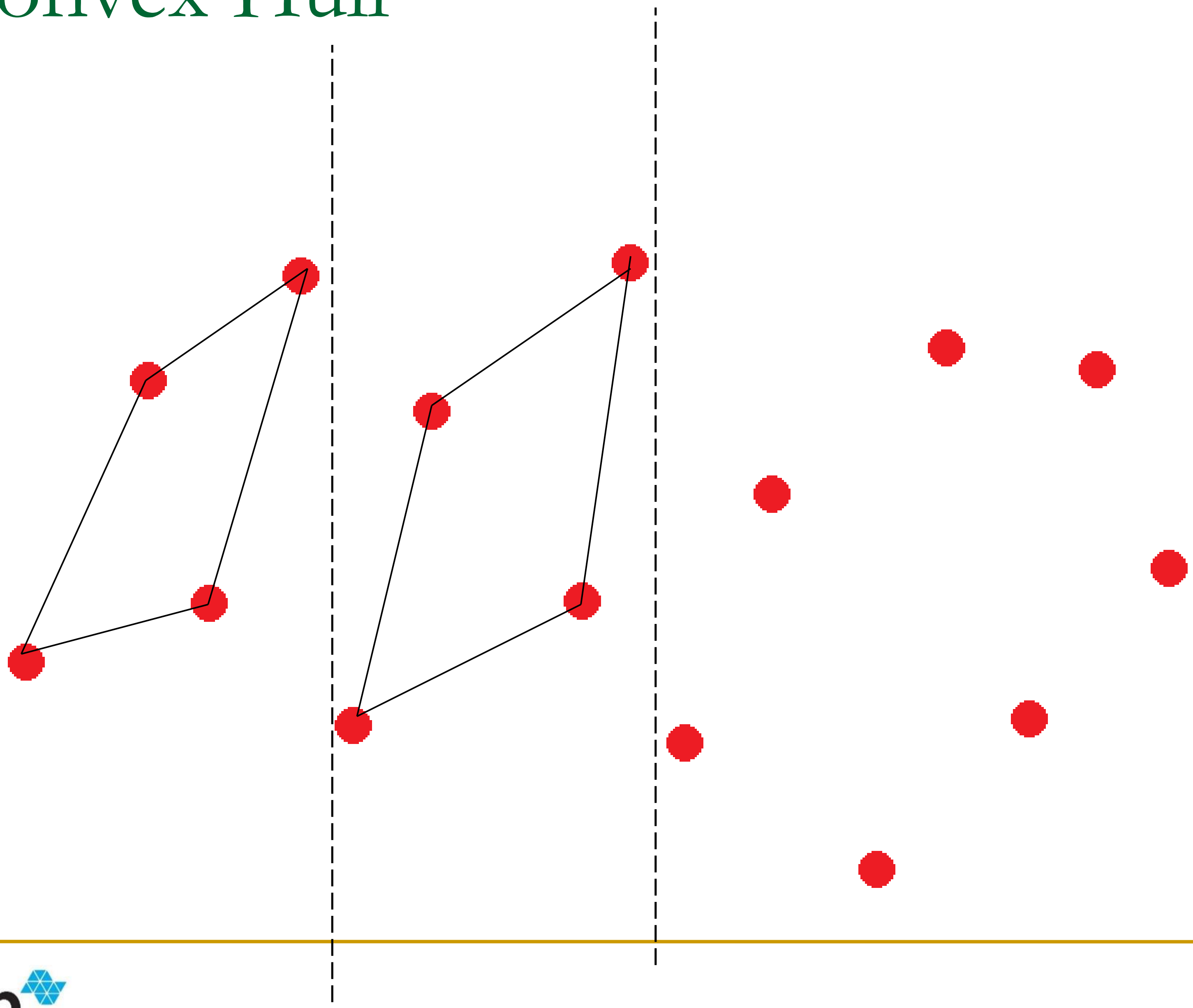
Convex Hull



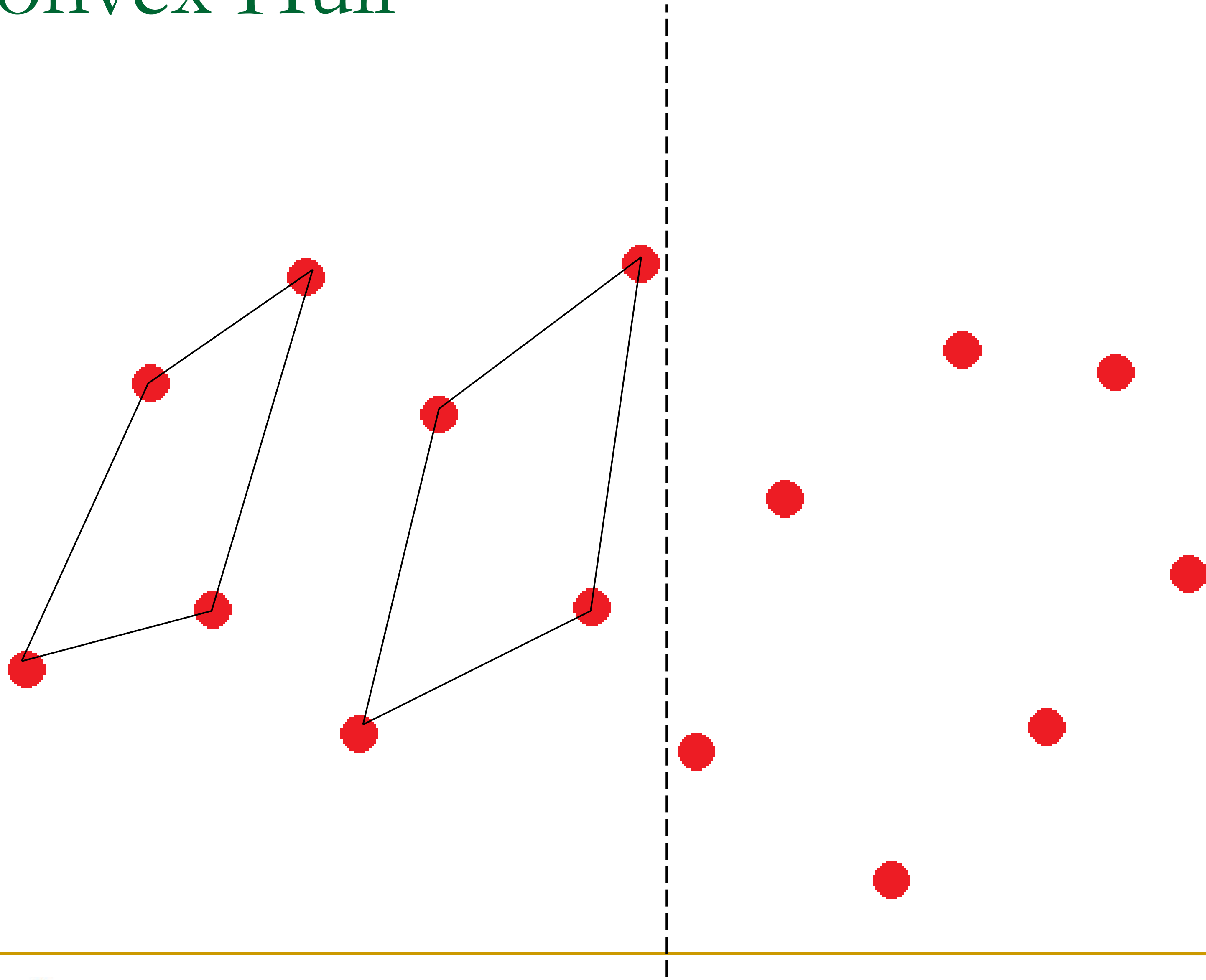
Convex Hull



Convex Hull

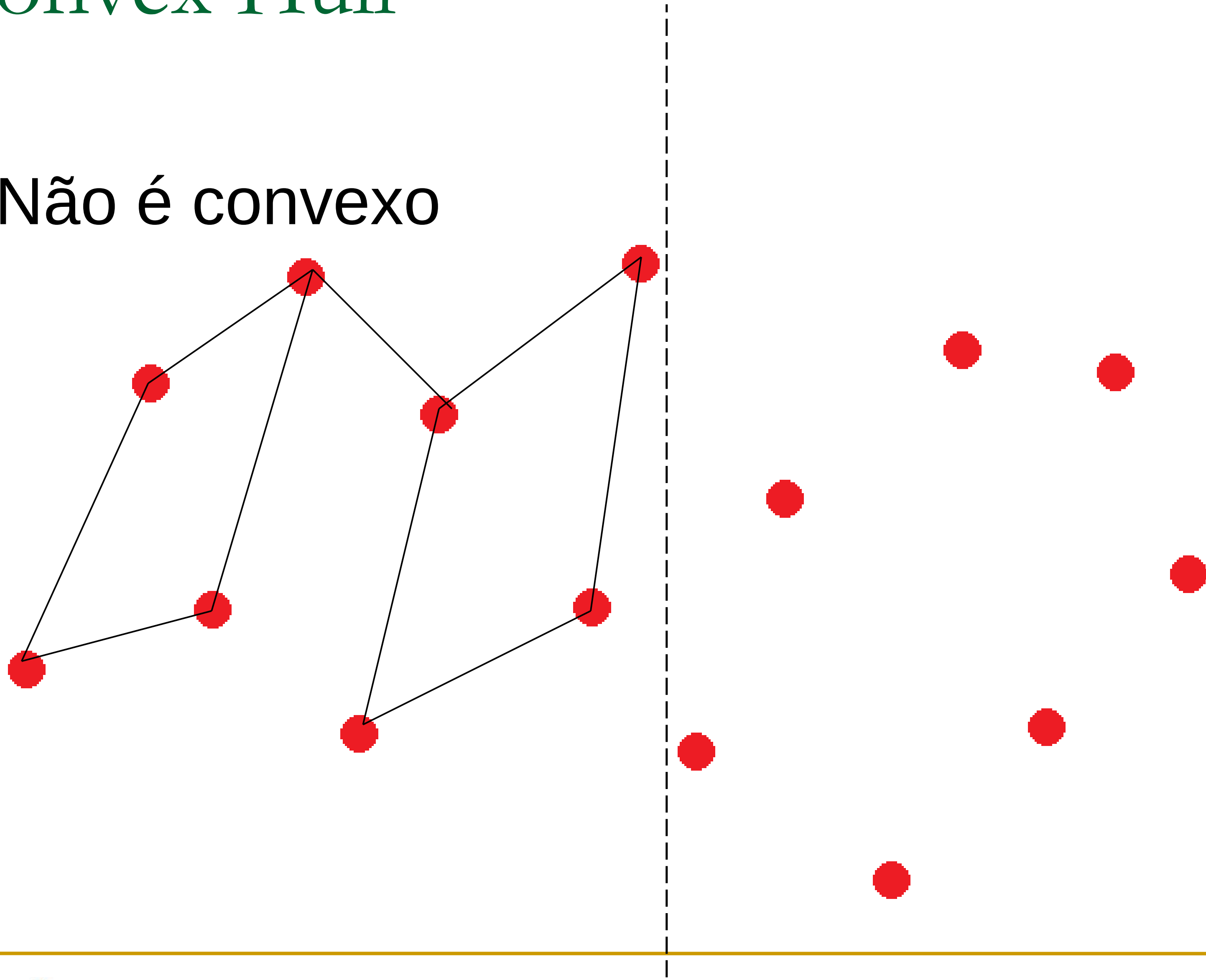


Convex Hull



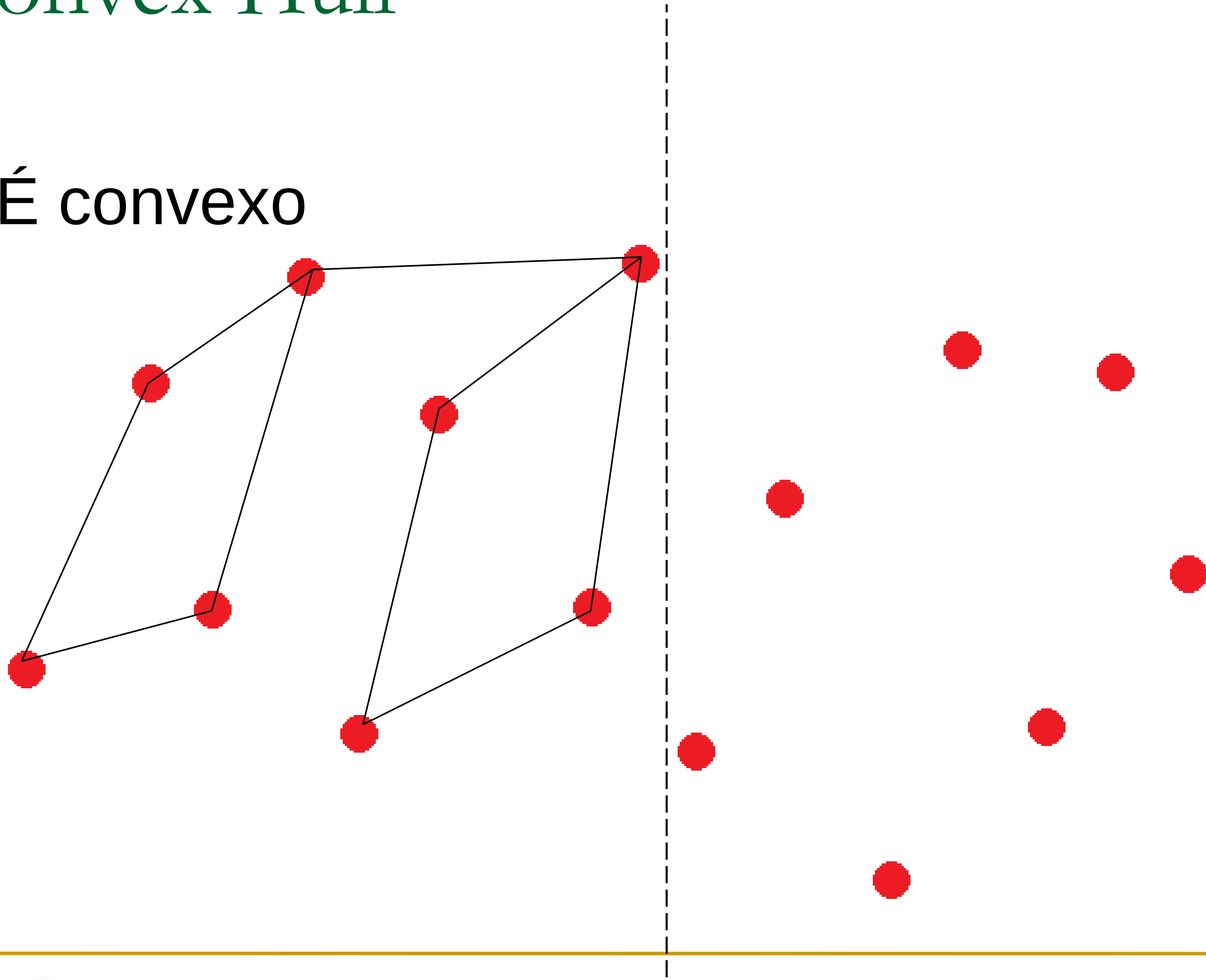
Convex Hull

- Não é convexo

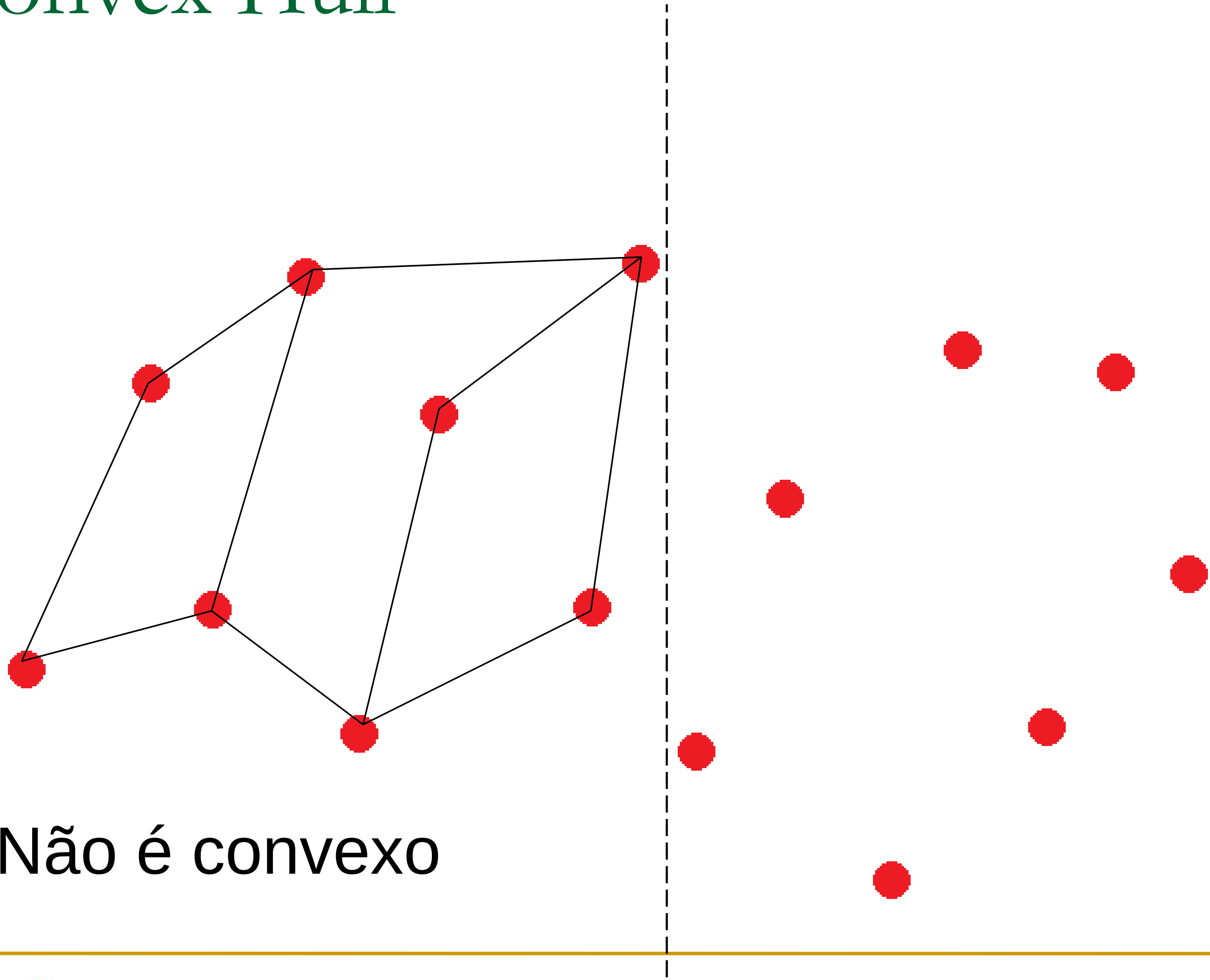


Convex Hull

- É convexo

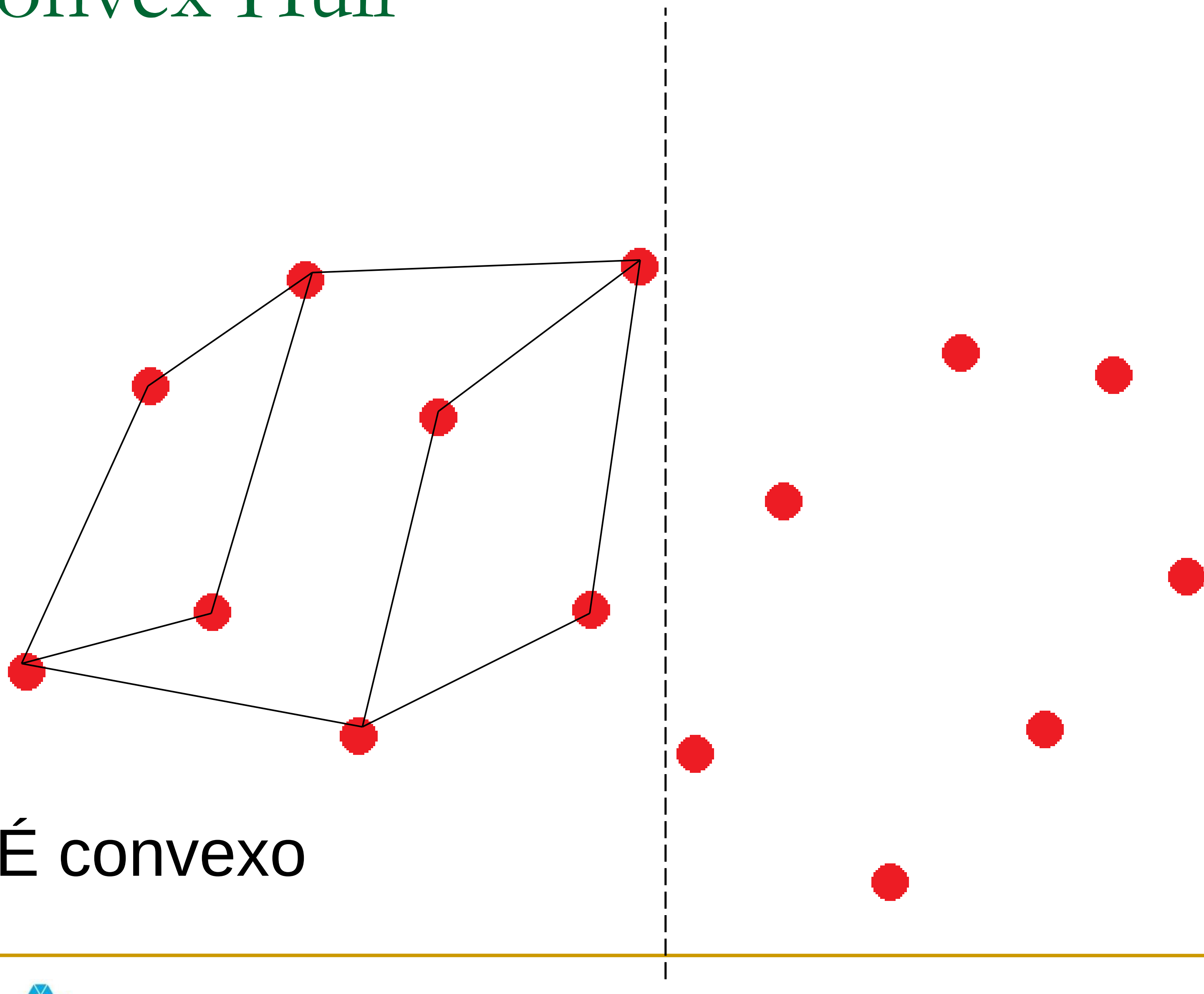


Convex Hull



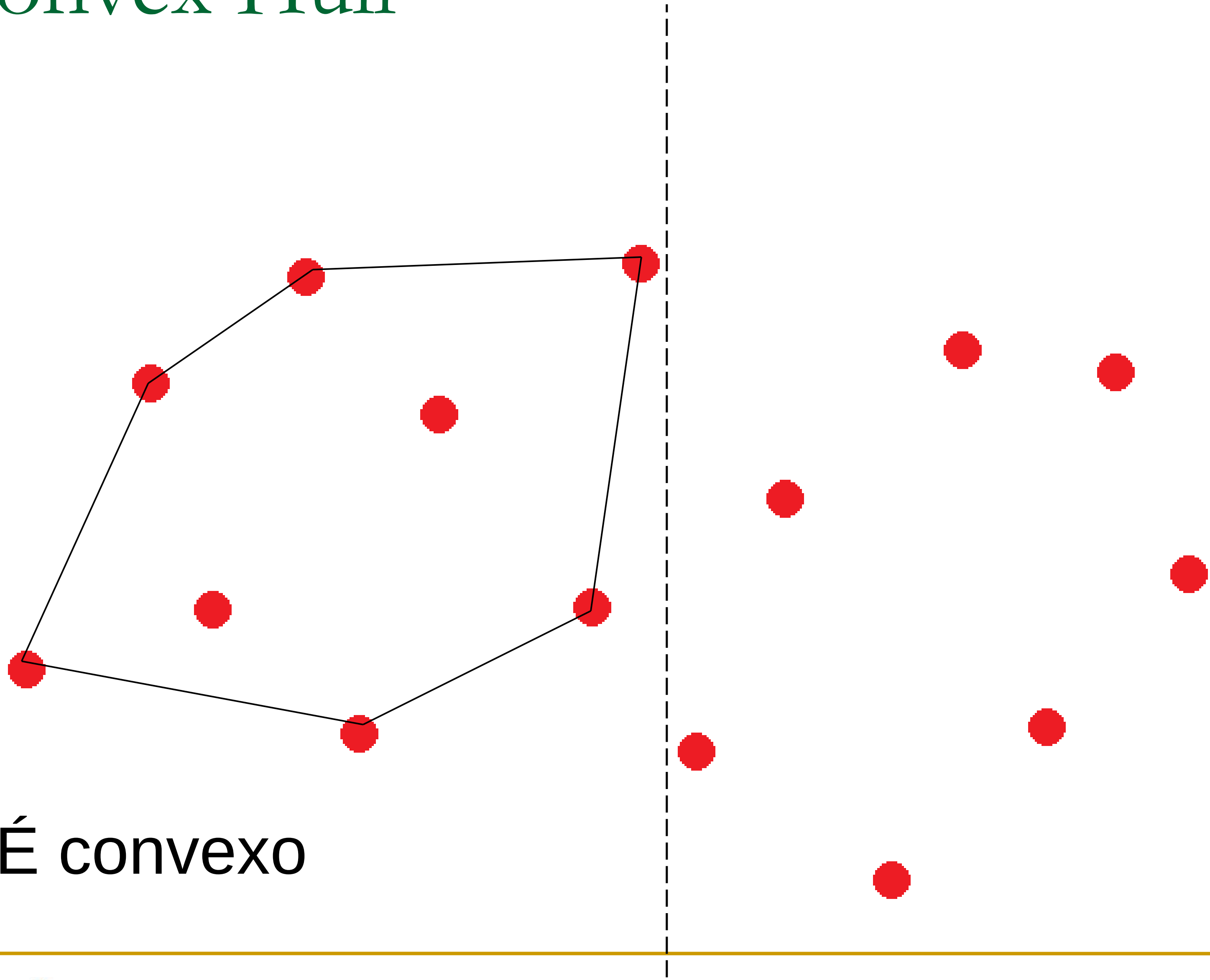
- Não é convexo

Convex Hull



- É convexo

Convex Hull

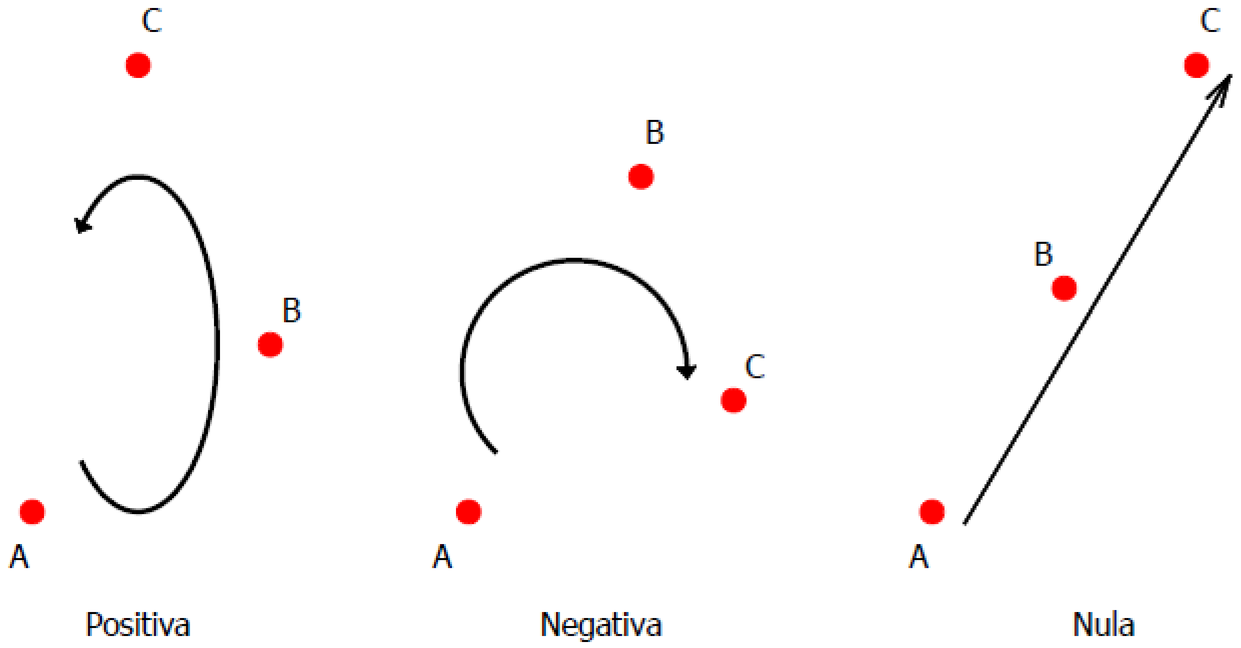


- É convexo

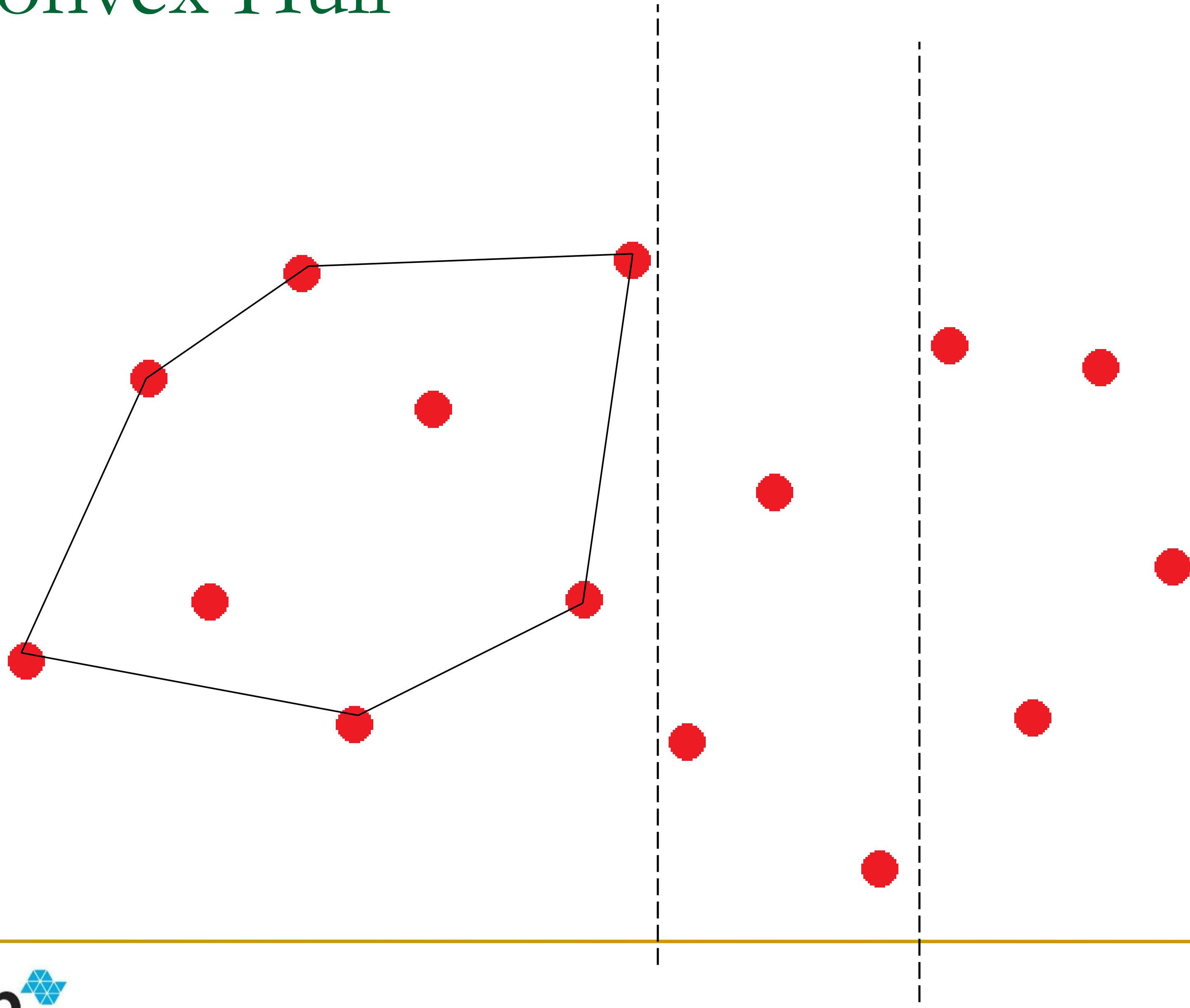
Convex Hull

- É possível identificar as orientações dos pontos de uma tripla (A, B, C)

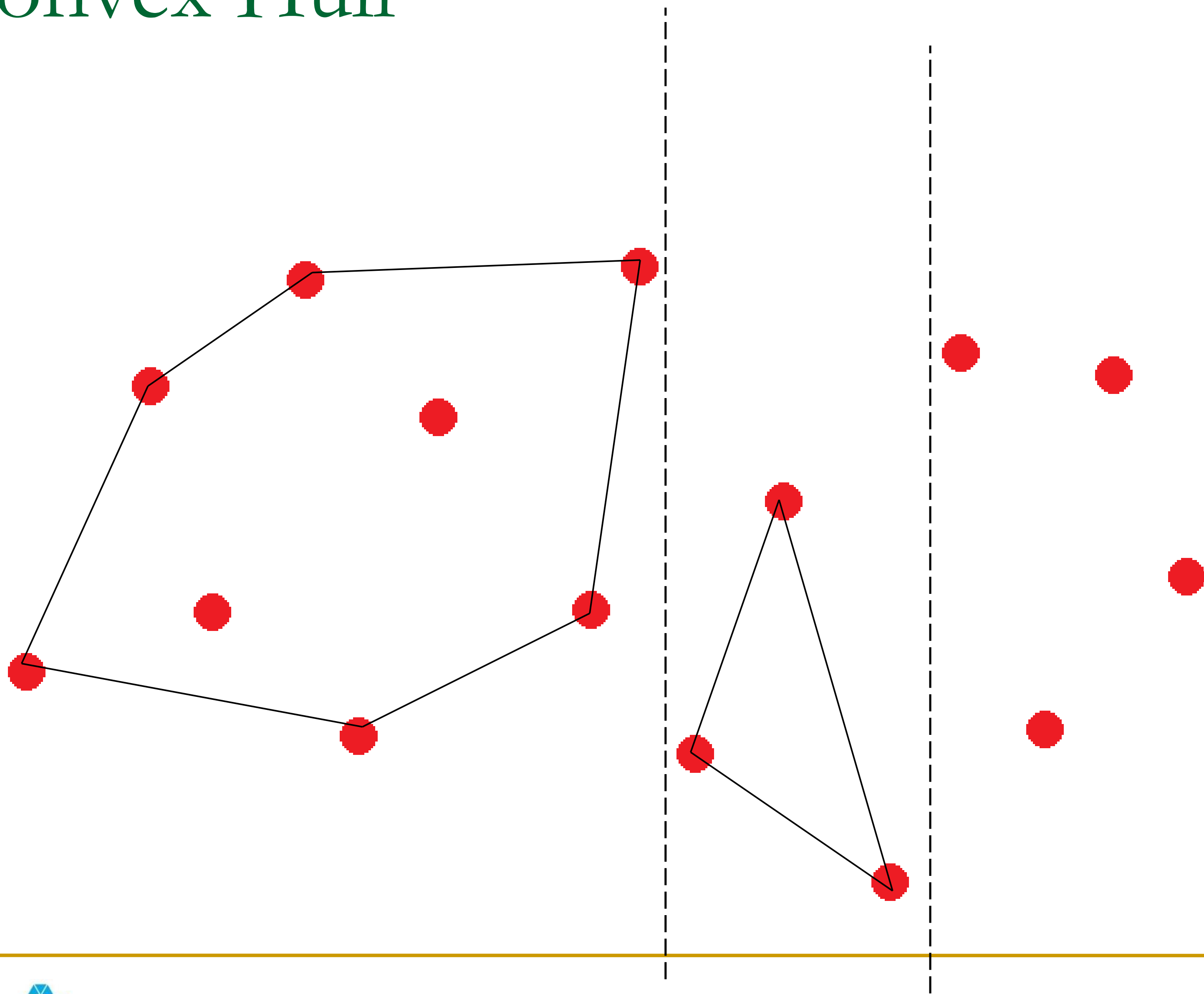
$$Orientacao(A,B,C) = \begin{vmatrix} 1 & A_x & A_y \\ 1 & B_x & B_y \\ 1 & C_x & C_y \end{vmatrix}$$



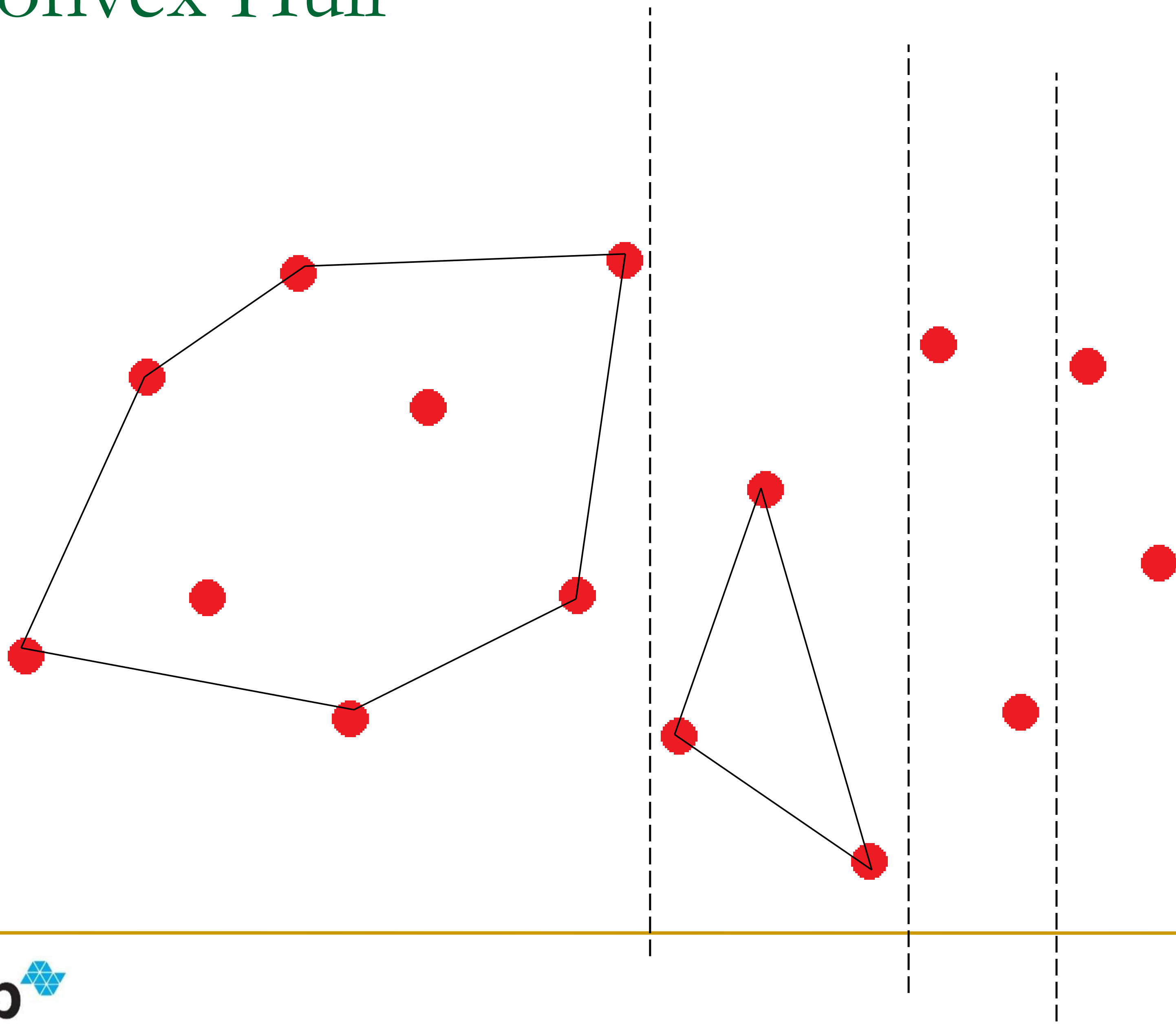
Convex Hull



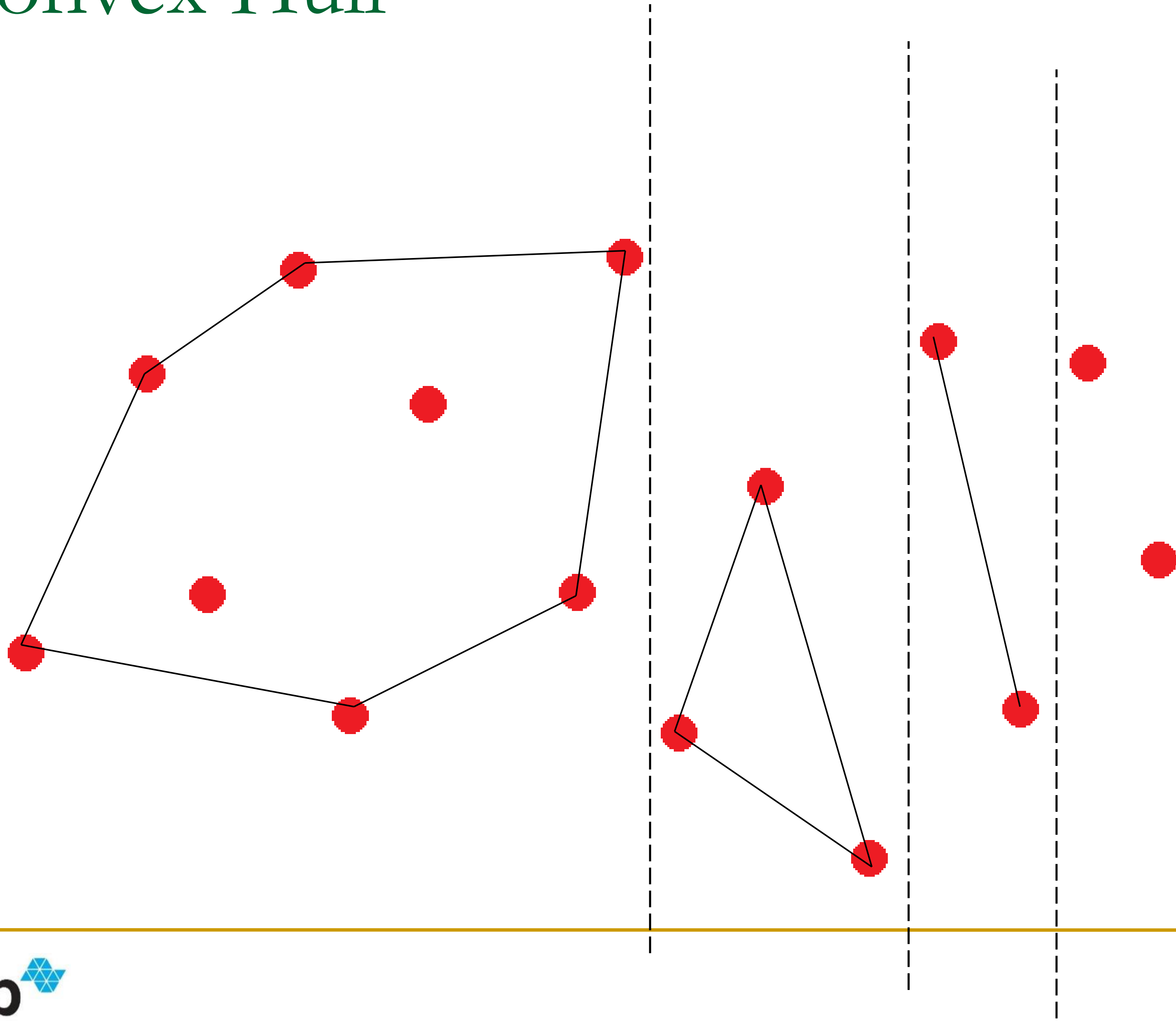
Convex Hull



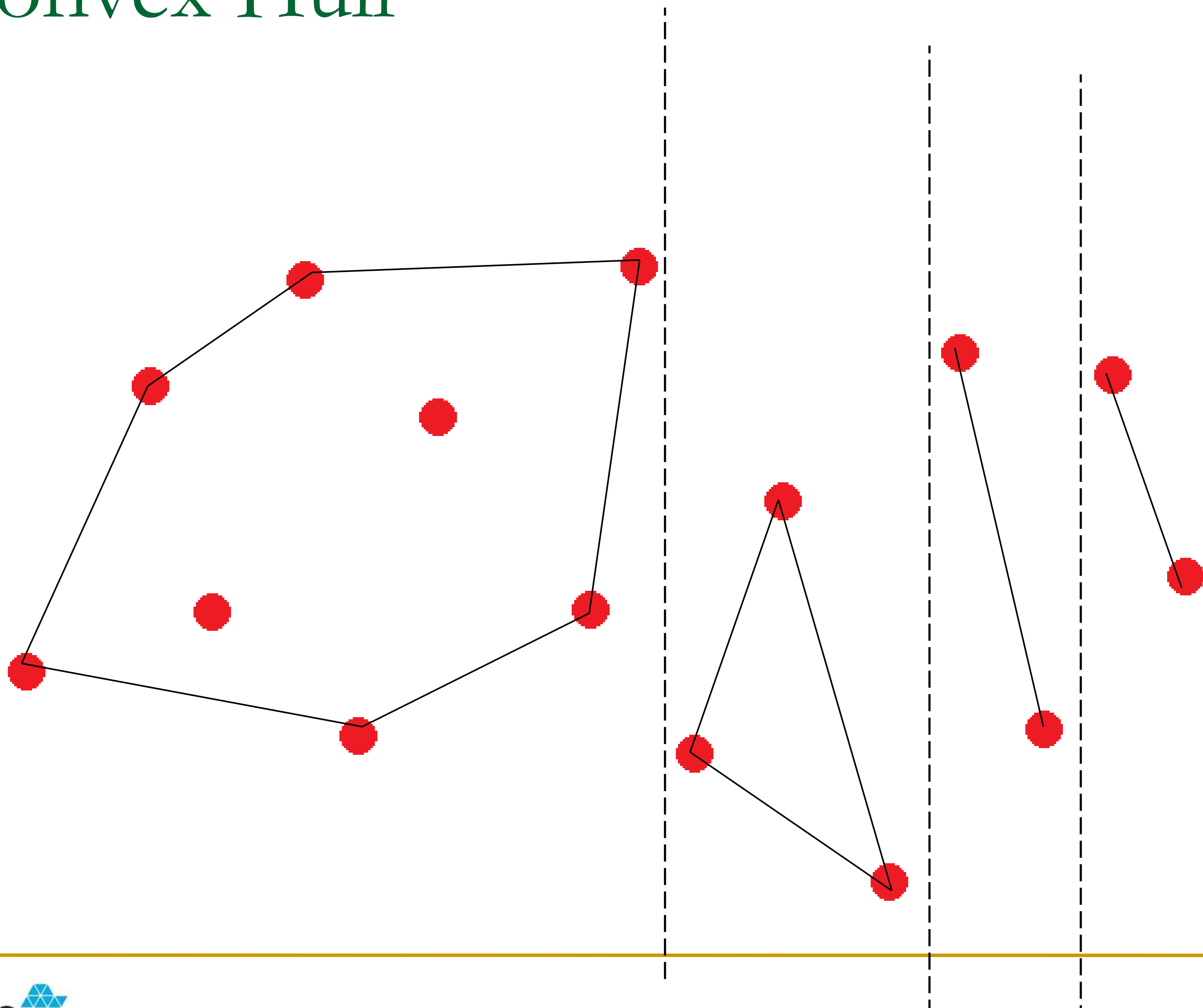
Convex Hull



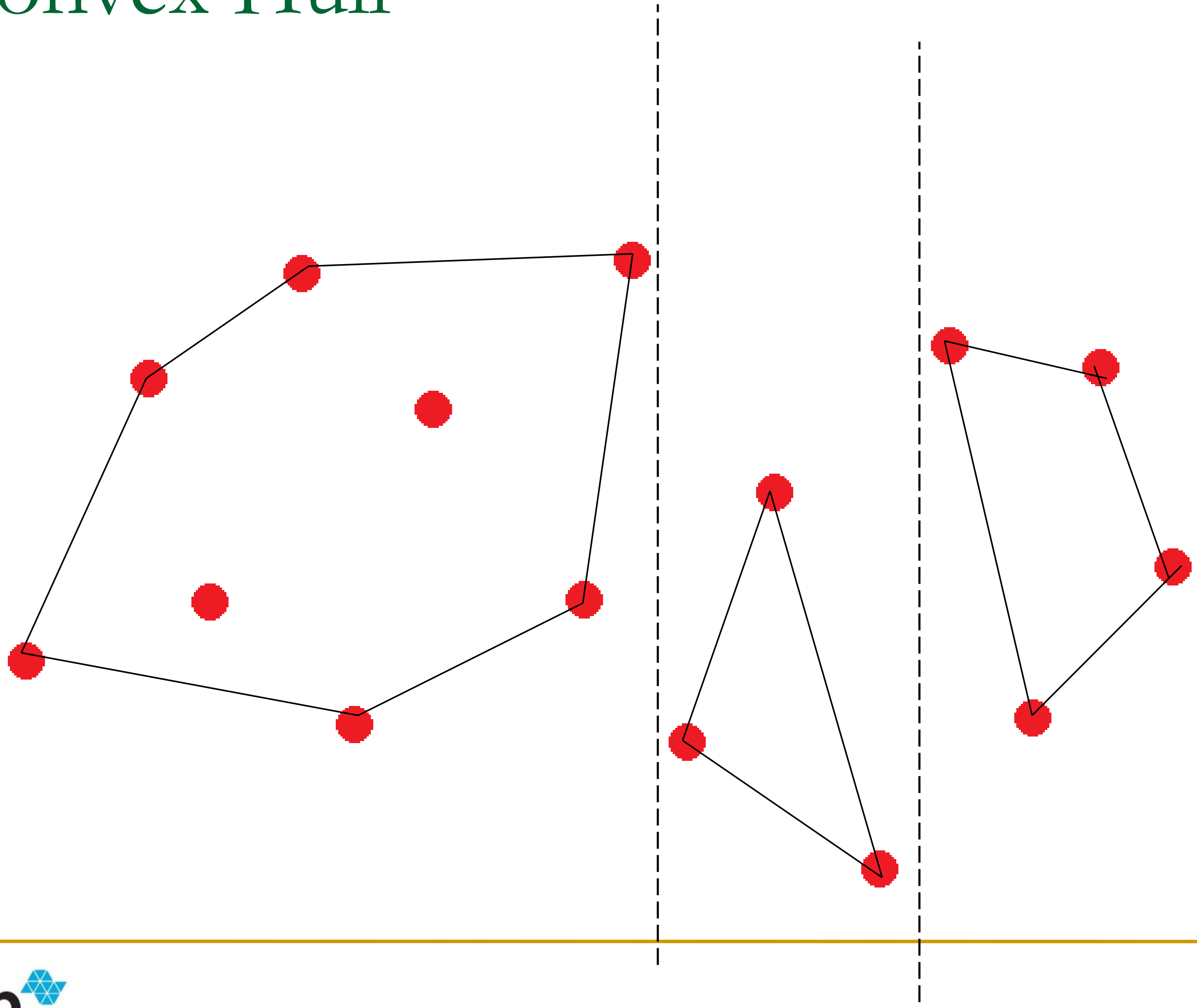
Convex Hull



Convex Hull

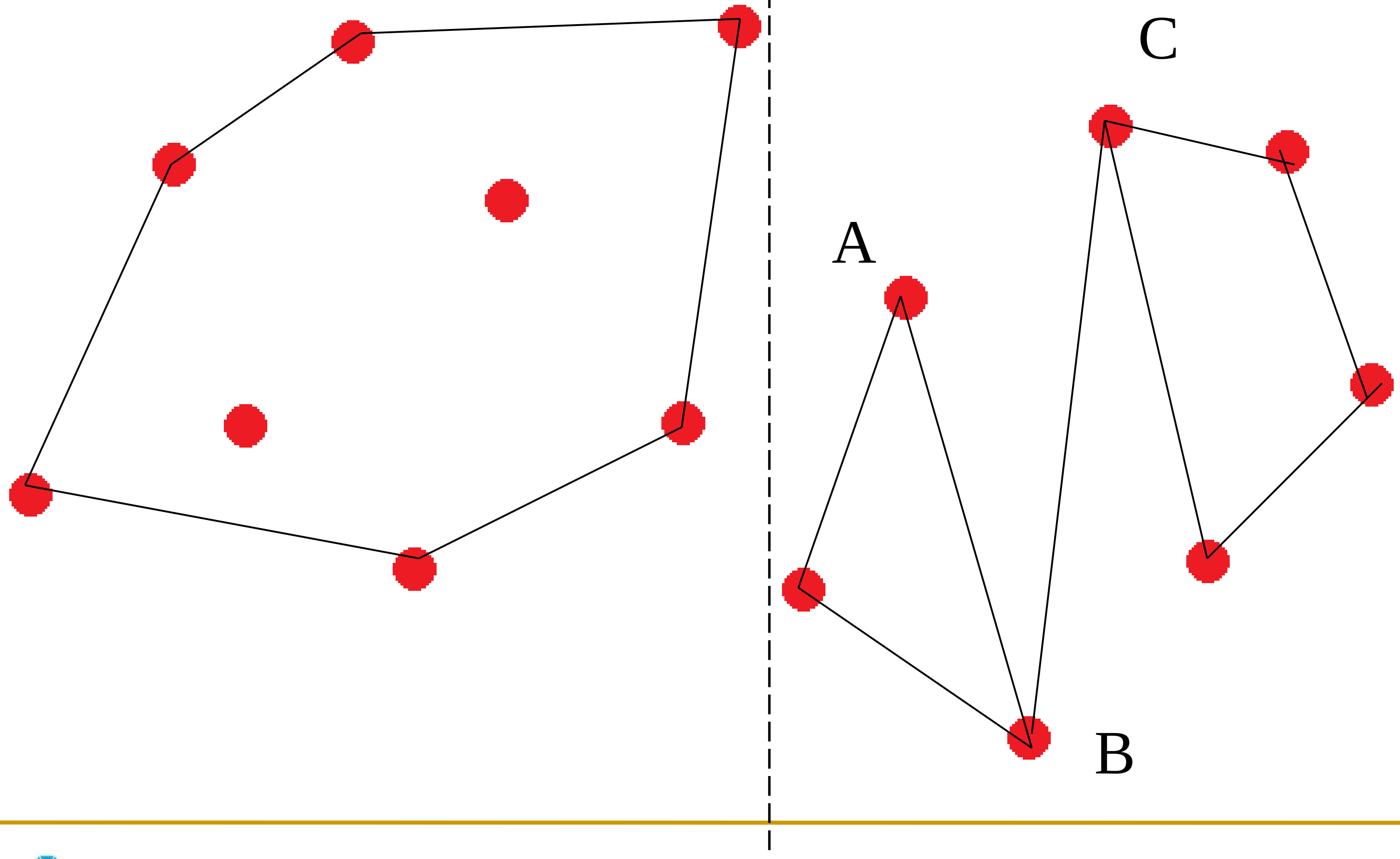


Convex Hull



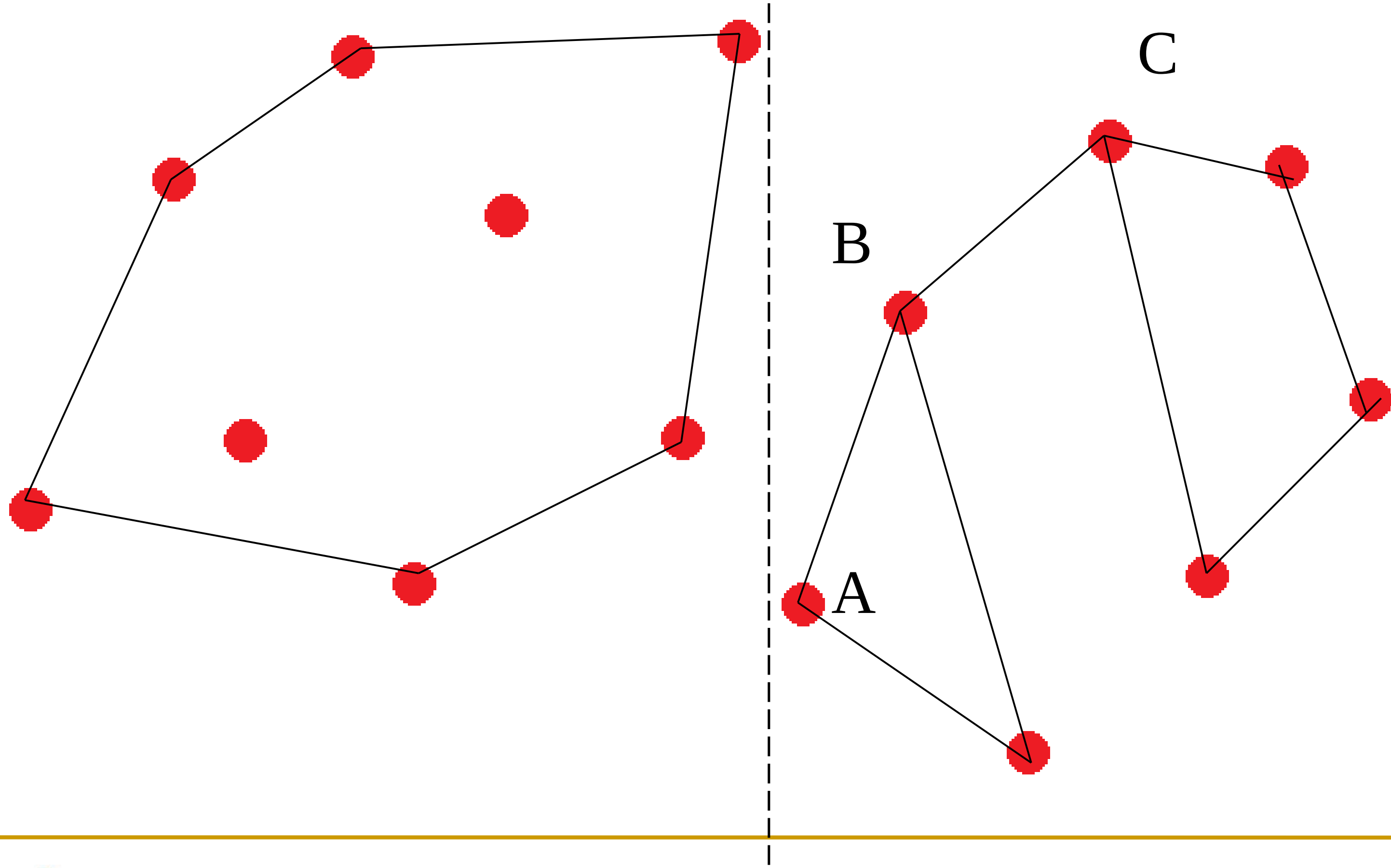
Convex Hull

- Tangente superior
- Sentido Anti-horário
- Não é convexo



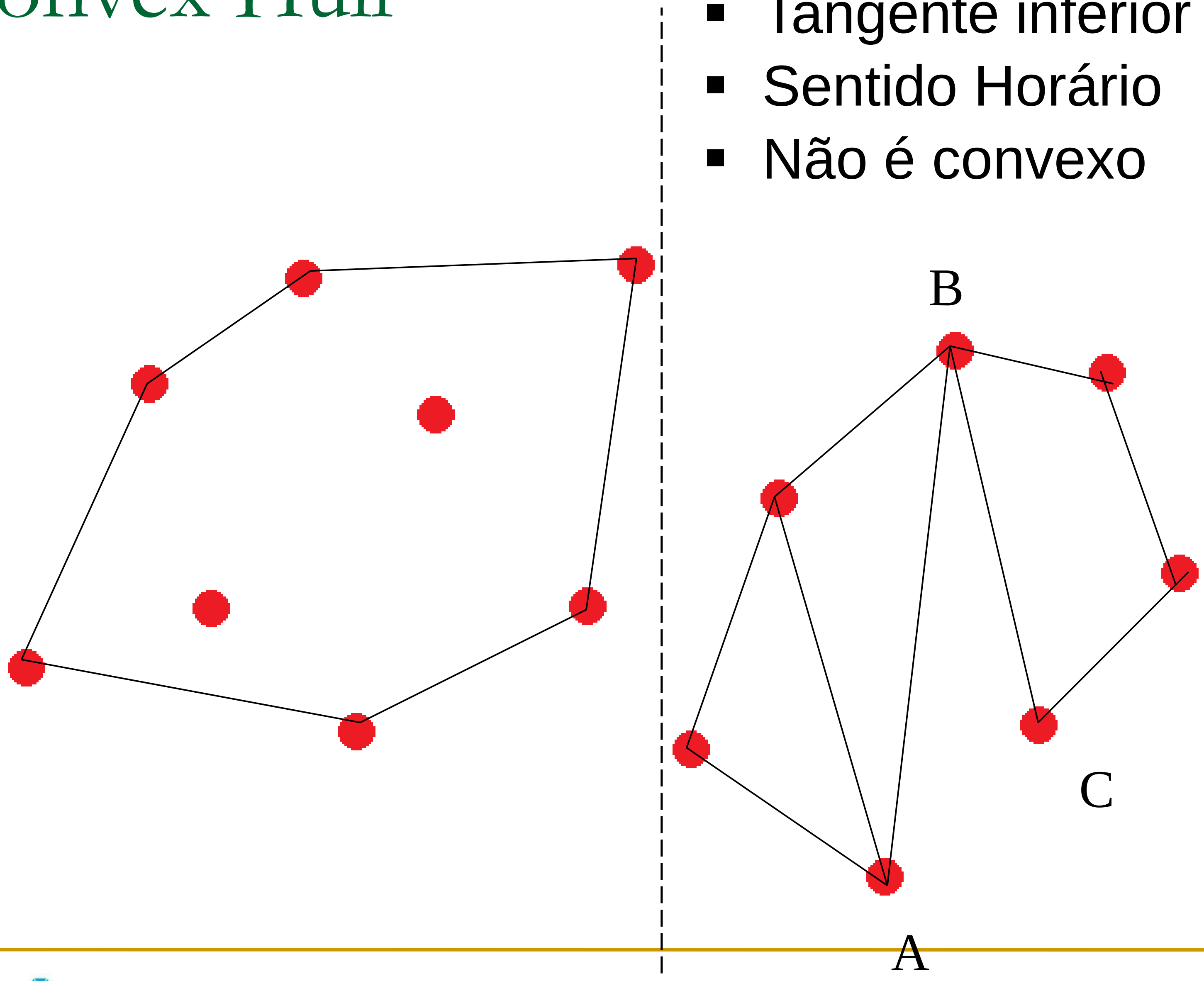
Convex Hull

- Tangente superior
- Sentido Horário
- É convexo



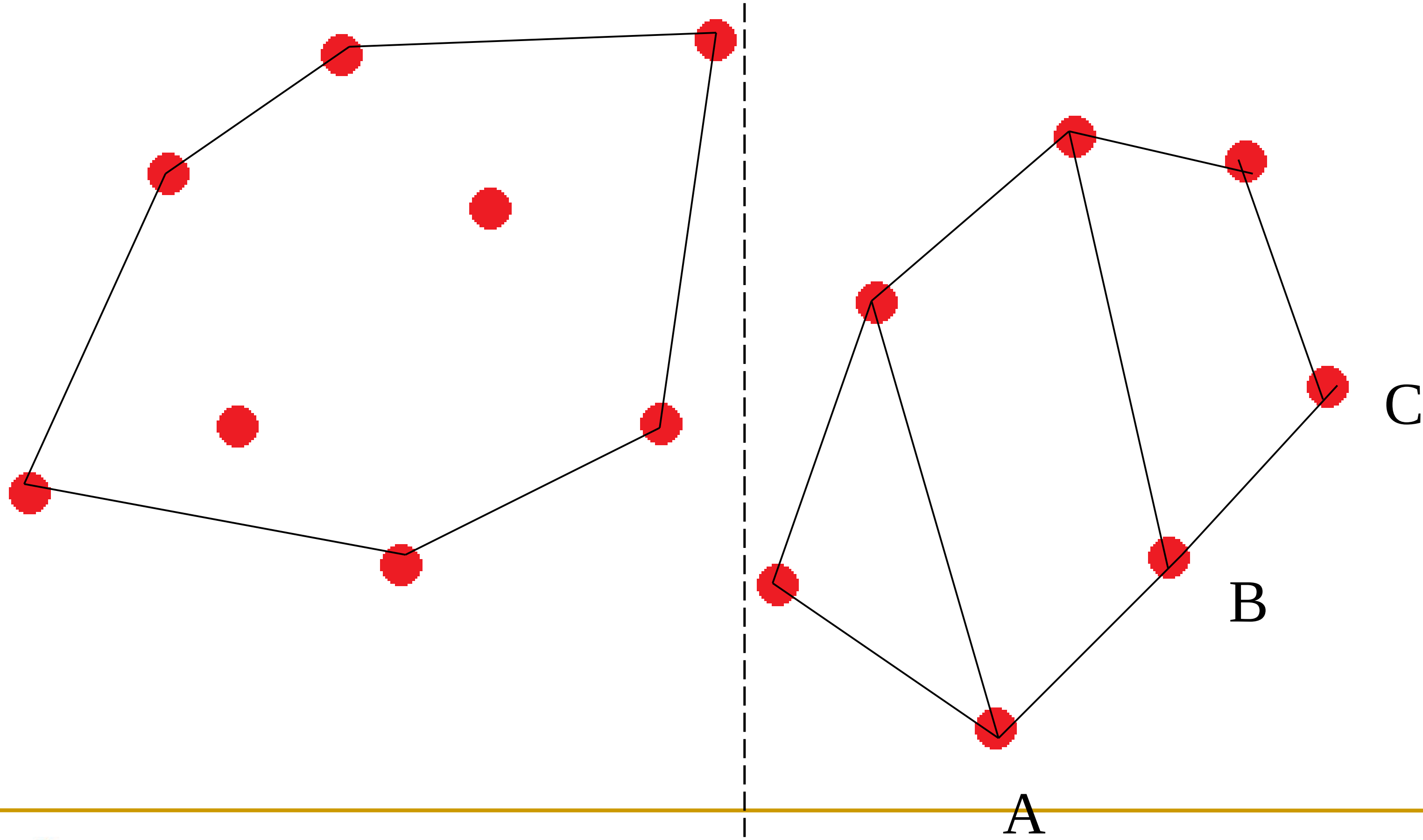
Convex Hull

- Tangente inferior
- Sentido Horário
- Não é convexo

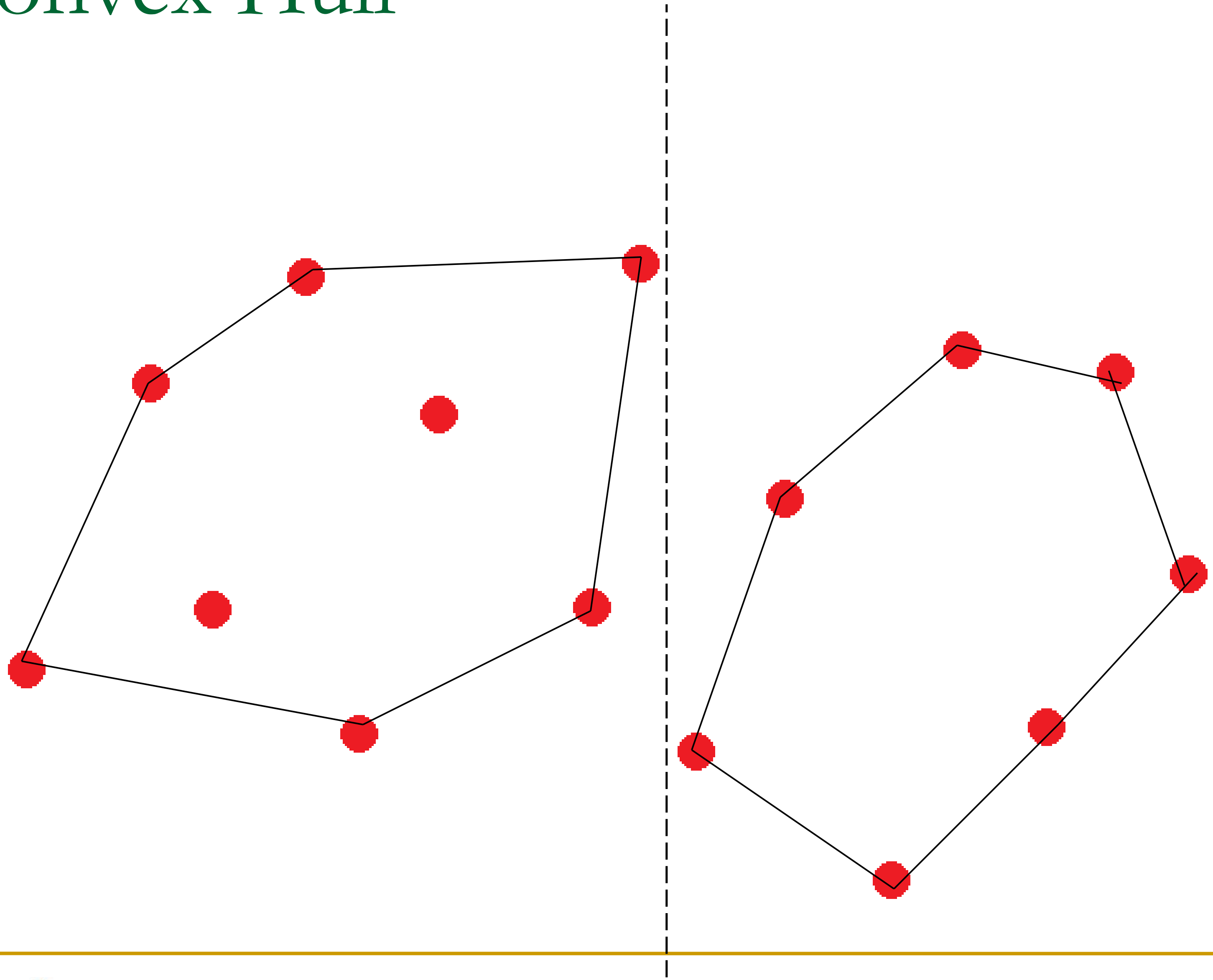


Convex Hull

- Tangente inferior
- Sentido Anti-Horário
- É convexo

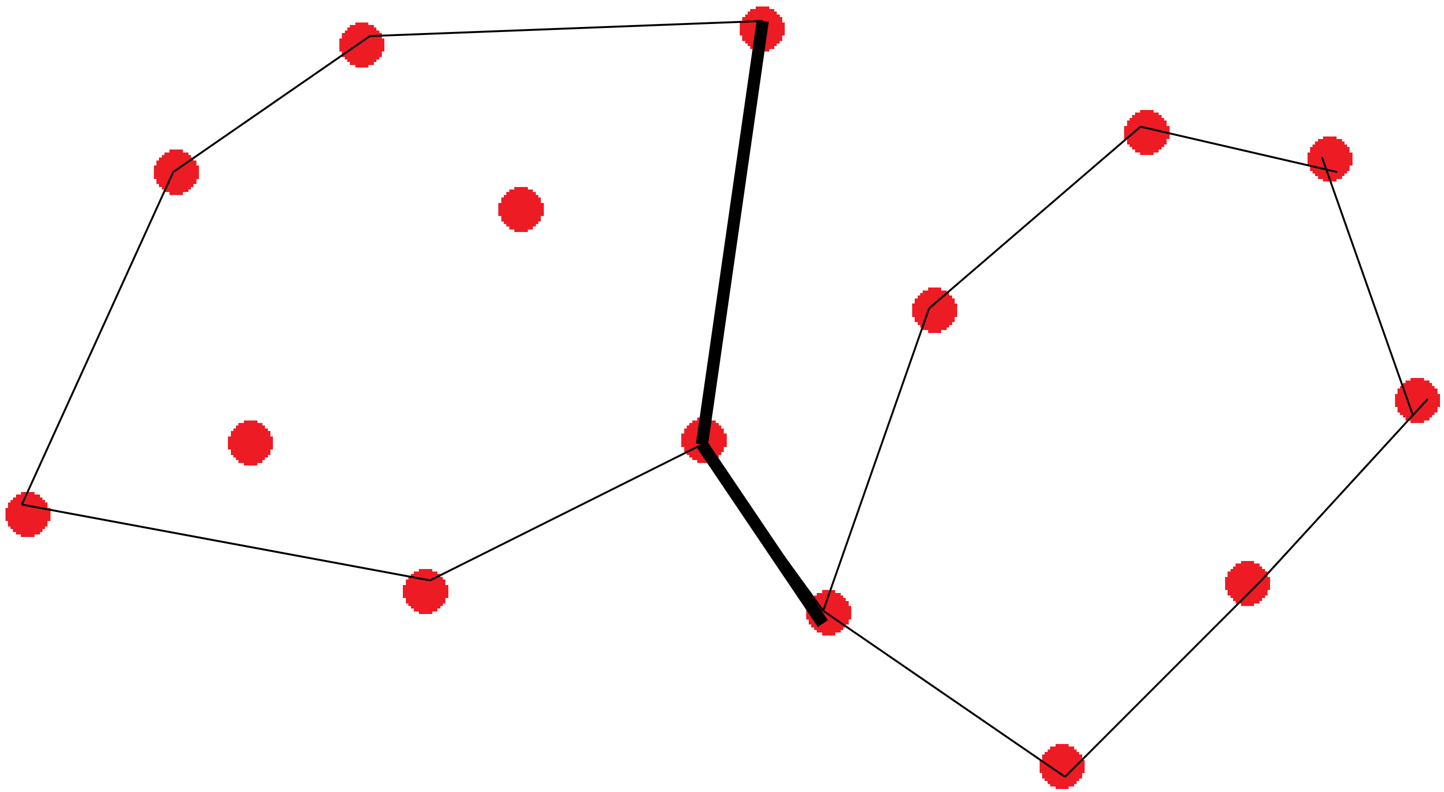


Convex Hull



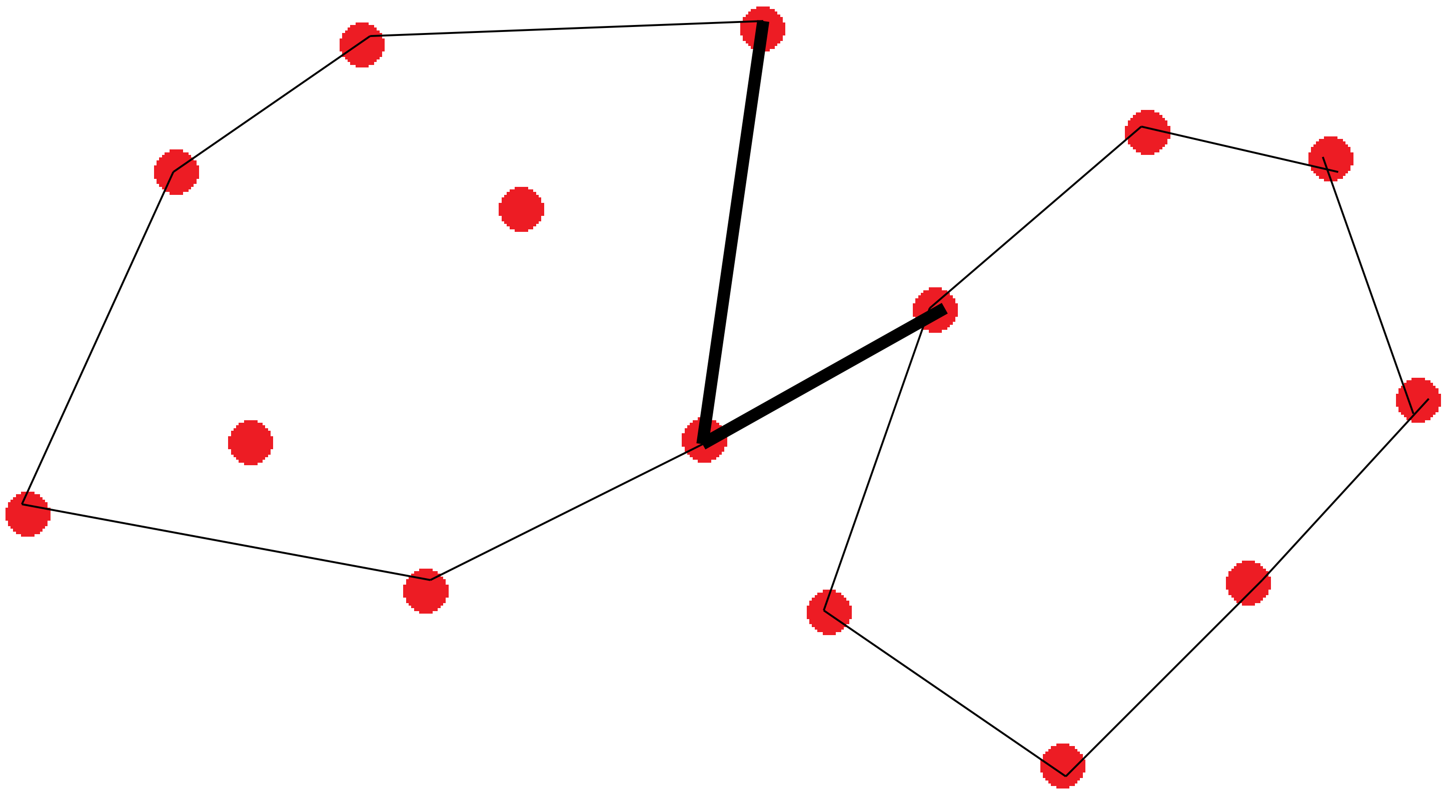
Convex Hull

- Não é convexo



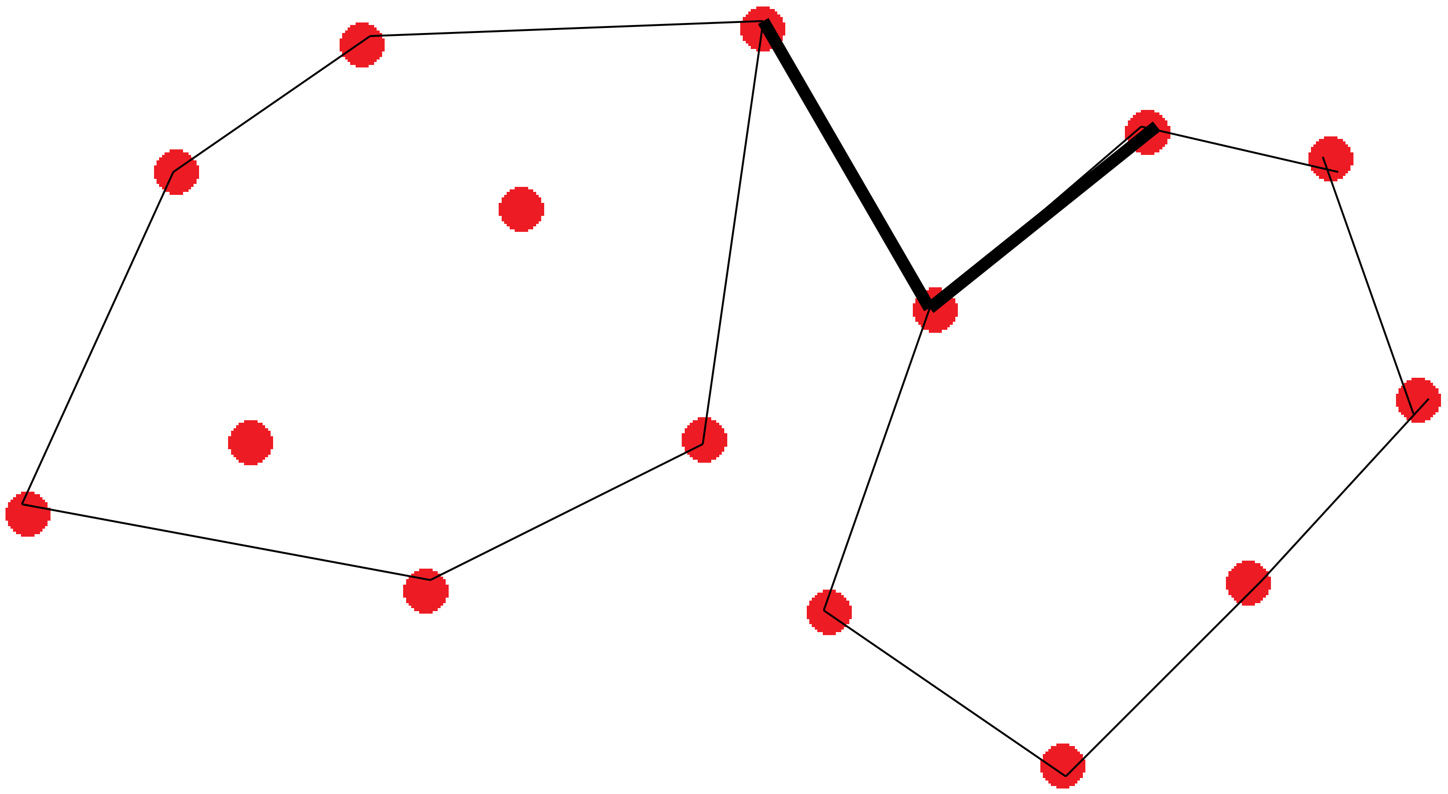
Convex Hull

- Não é convexo



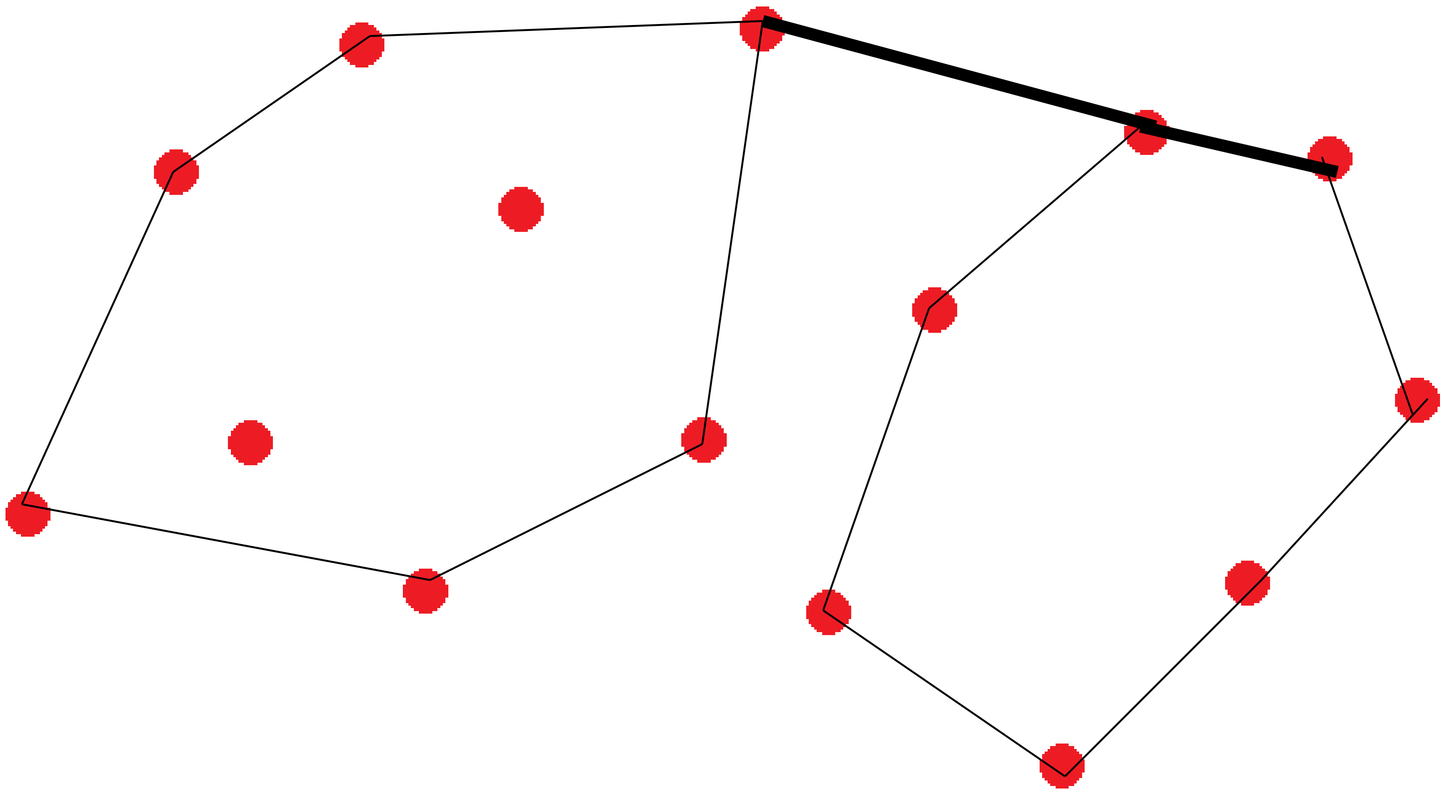
Convex Hull

- Não é convexo



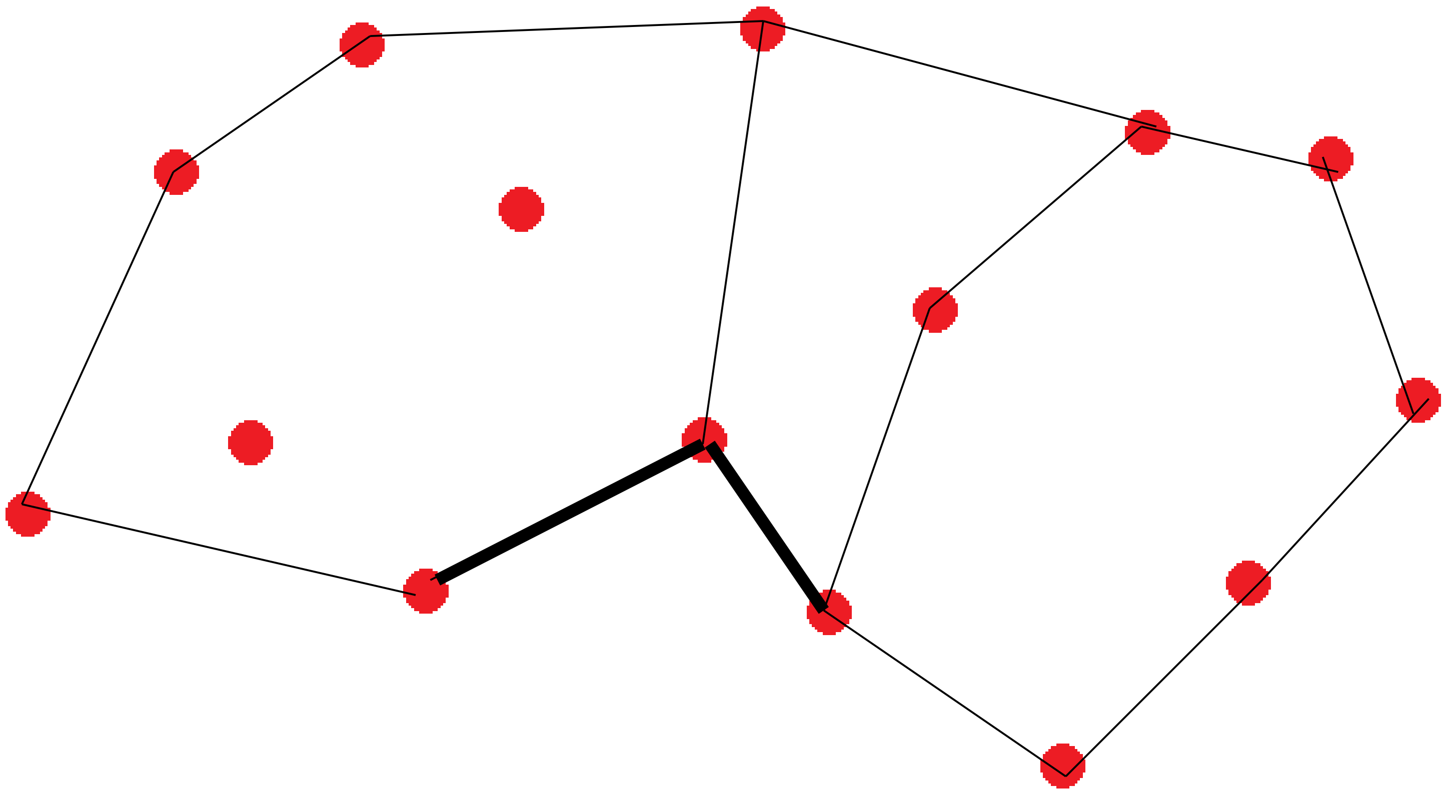
Convex Hull

- É convexo



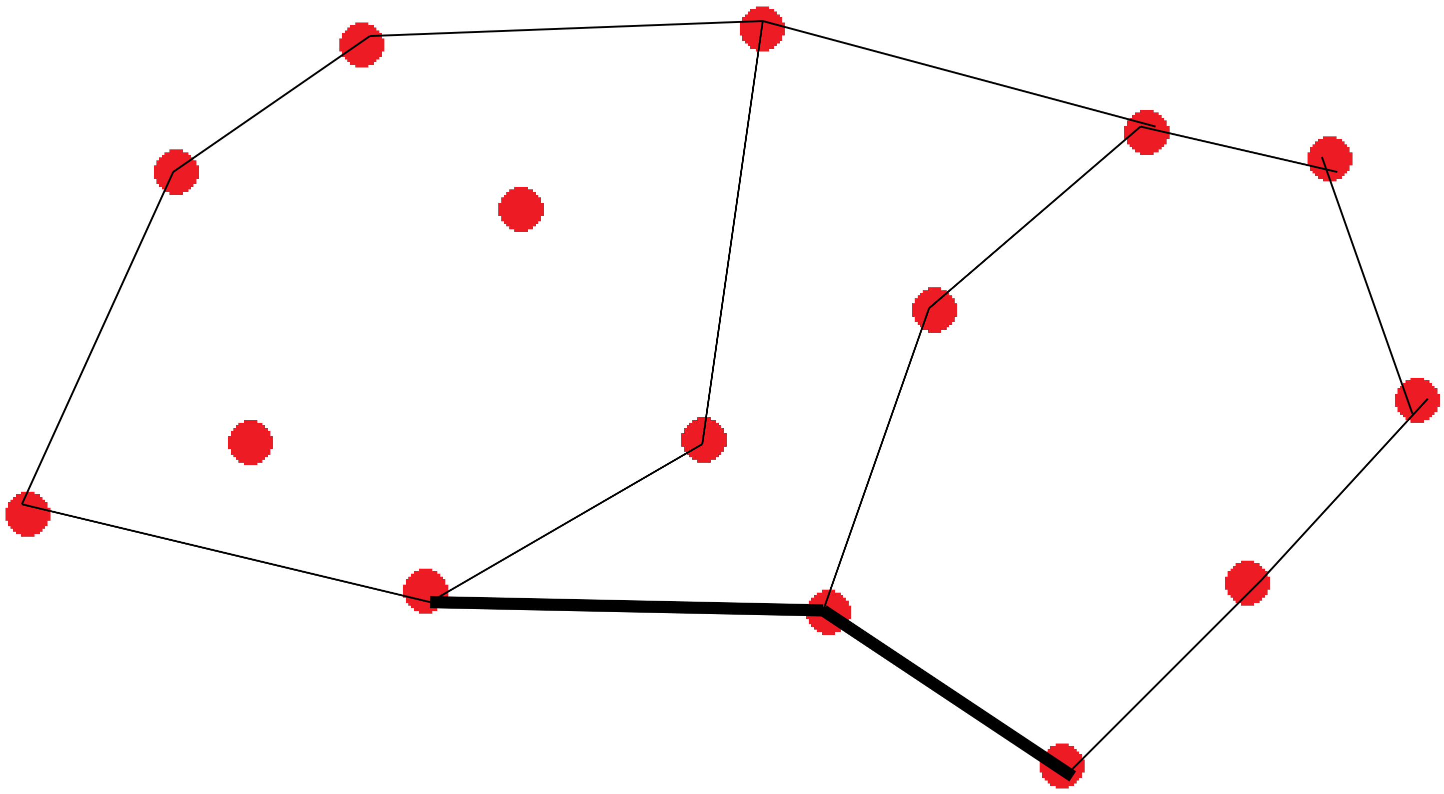
Convex Hull

- Não é convexo



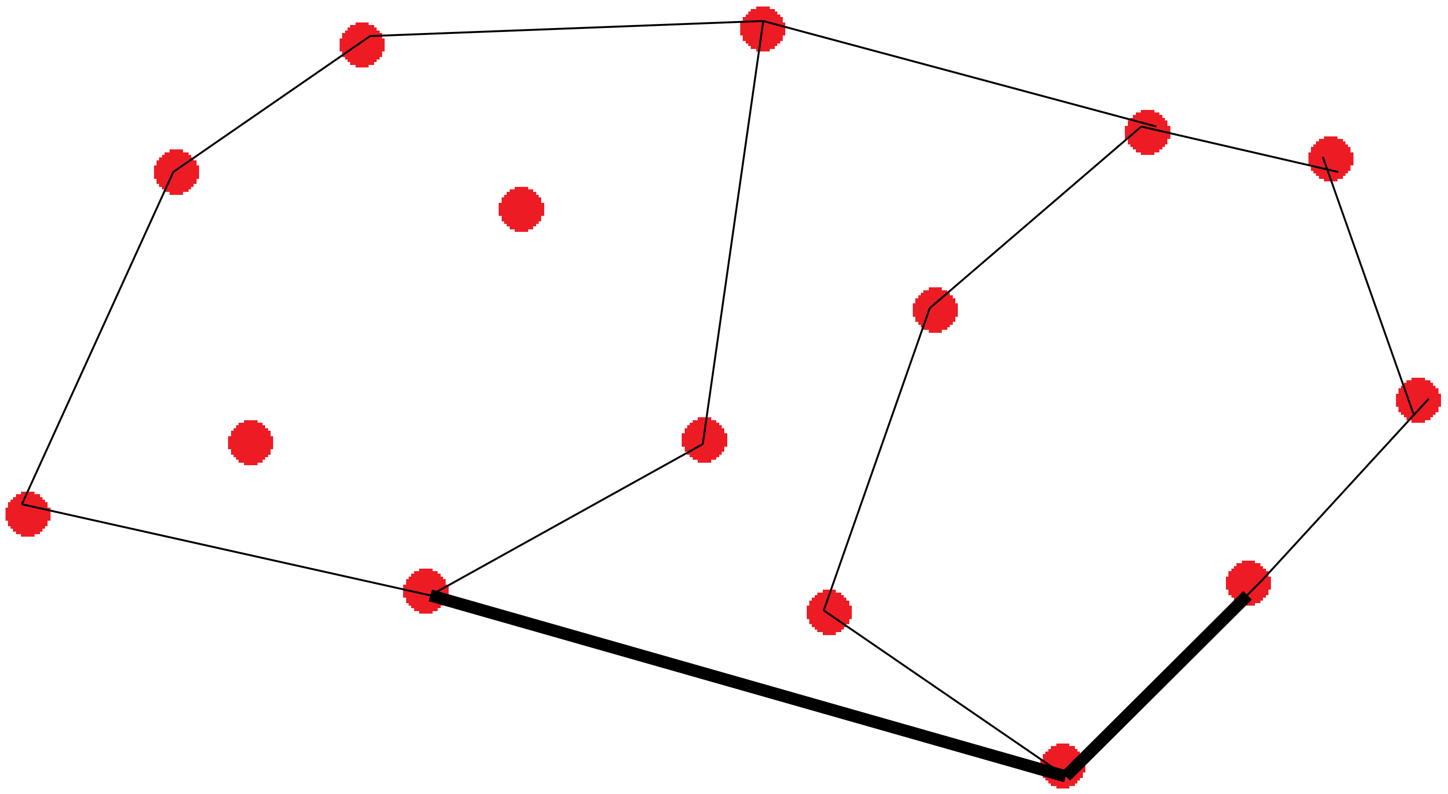
Convex Hull

- Não é convexo



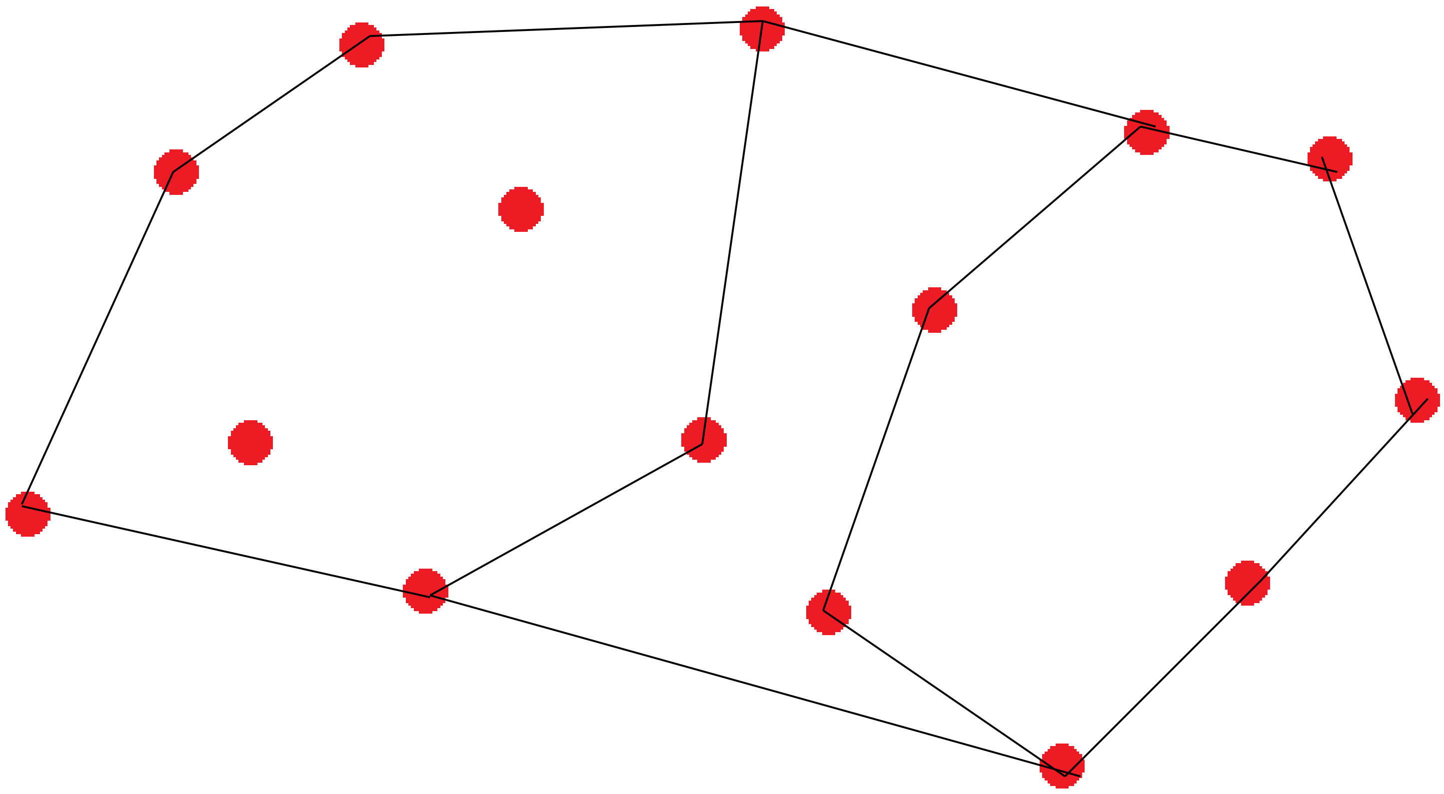
Convex Hull

- É convexo

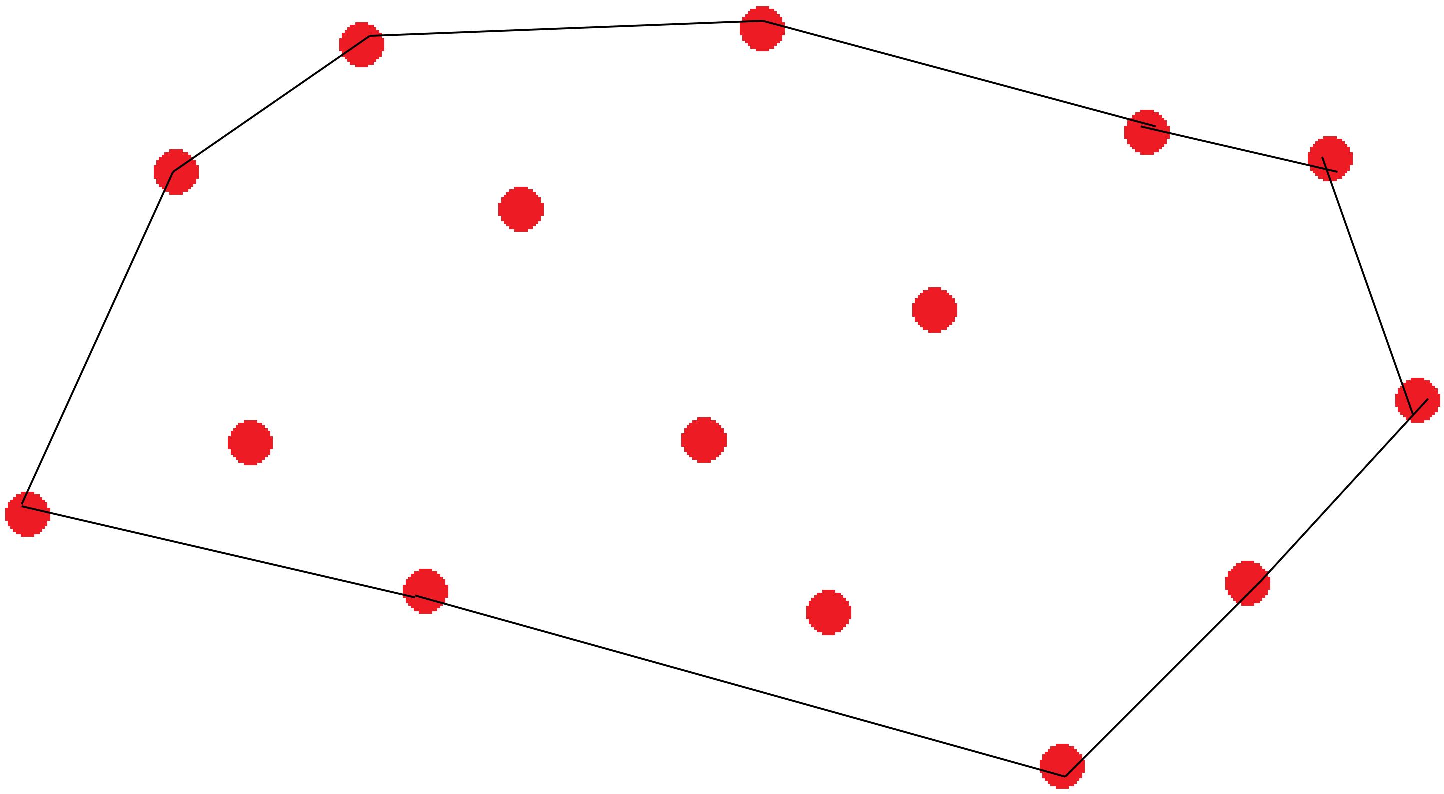


Convex Hull

- É convexo



Convex Hull



Parte inicial deste material foi adaptado de

- Professor Moacir Ponti Junior
  - ICMC/USP – São Carlos
  - Aula:
    - Projeto de Algoritmos: paradigmas
    - SCC201/501 - Introdução à Ciência de Computação II

Bibliografia

- LEVITIN, A. Introduction to The Design & Analysis of Algorithms, Addison Wesley, 2003, 497p.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; (2002). Algoritmos –Teoria e Prática. Tradução da 2ª edição americana. Rio de Janeiro. Editora Campus
- TAMASSIA, ROBERTO; GOODRICH, MICHAEL T. (2004). Projeto de Algoritmos -Fundamentos, Análise e Exemplos da Internet
- ZIVIANI, N. (2007). Projeto e Algoritmos com implementações em Java e C++. São Paulo. Editora Thomson

Bibliografia

BIBLIOGRAFIA BÁSICA

1. CORMEN, T. H.; LEISERSON C. E.; RIVEST, R. L.; STEIN, C. **Algoritmos**: teoria e prática, 2ª ed. Rio de Janeiro: Editora Campus, 2002. 916p.
2. SEDGEWICK, R. **Algorithms in C**: graph algorithms, 3ª ed. Part 5, Addison Wesley, 2002. 482p.
3. NETTO, P. O. B. **Grafos**: teoria, modelos, algoritmos. 2ª ed. São Paulo: Editora Edgard Blücher Ltda, 2001. 304p.
4. SZWARCFITER, J. L. **Grafos e Algoritmos Computacionais**. Rio de Janeiro: Editora Campus, 1984. 216p.
5. ZIVIANI, N. **Projeto de algoritmos**: com implementações em PASCAL e C. 2ª ed. São Paulo: Pioneira Thomson Learning Ltda, 2004. 552p.

BIBLIOGRAFIA COMPLEMENTAR

1. DROZDEK, A. **Estrutura de Dados e Algoritmos em C++**. São Paulo: Pioneira Thomson Learning Ltda, 2002. 579p.
2. GOODRICH, M. T.; TAMASSIA, R. **Estruturas de Dados e Algoritmos em JAVA**. 2ª ed. Bookman Companhia Editora, 2002. 584p.
3. SCHEINERMAN, E. R. **Matemática discreta**: uma introdução. São Paulo: Pioneira Thomson Learning, 2003. 532 p.
4. WILSON, R. J.; WATKINS, J. J. **Graphs**: an introductory approach. New York: John Wiley & Sons, Inc, 1990. 340p.