



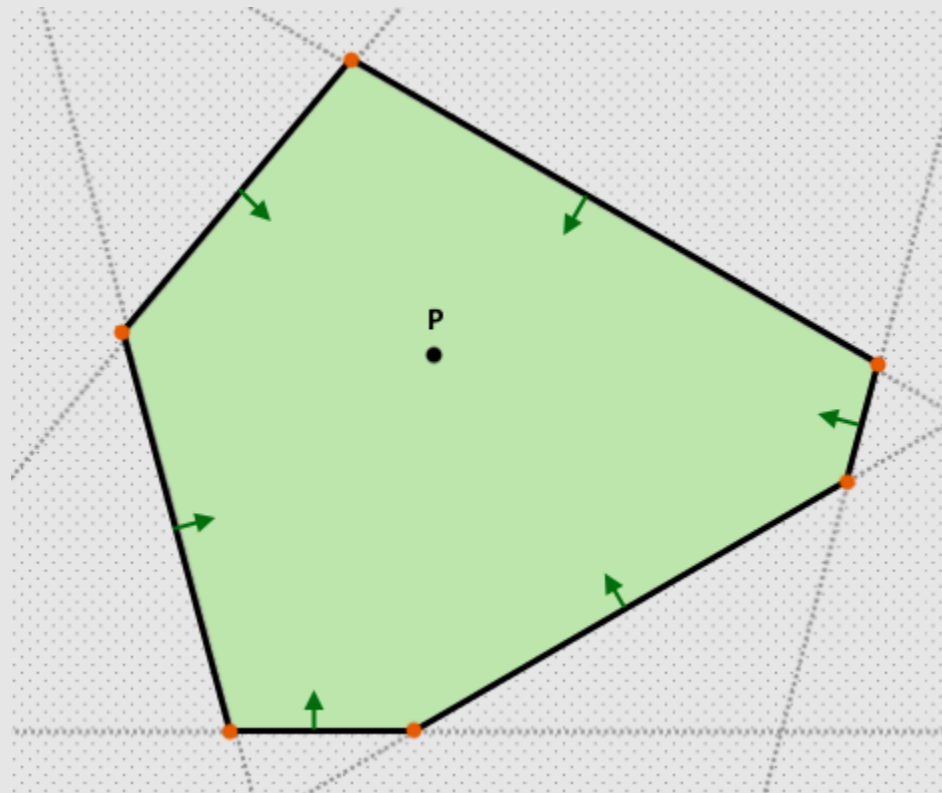
Convex Hull

Uma abordagem interativa ao problema de fecho convexo



Base

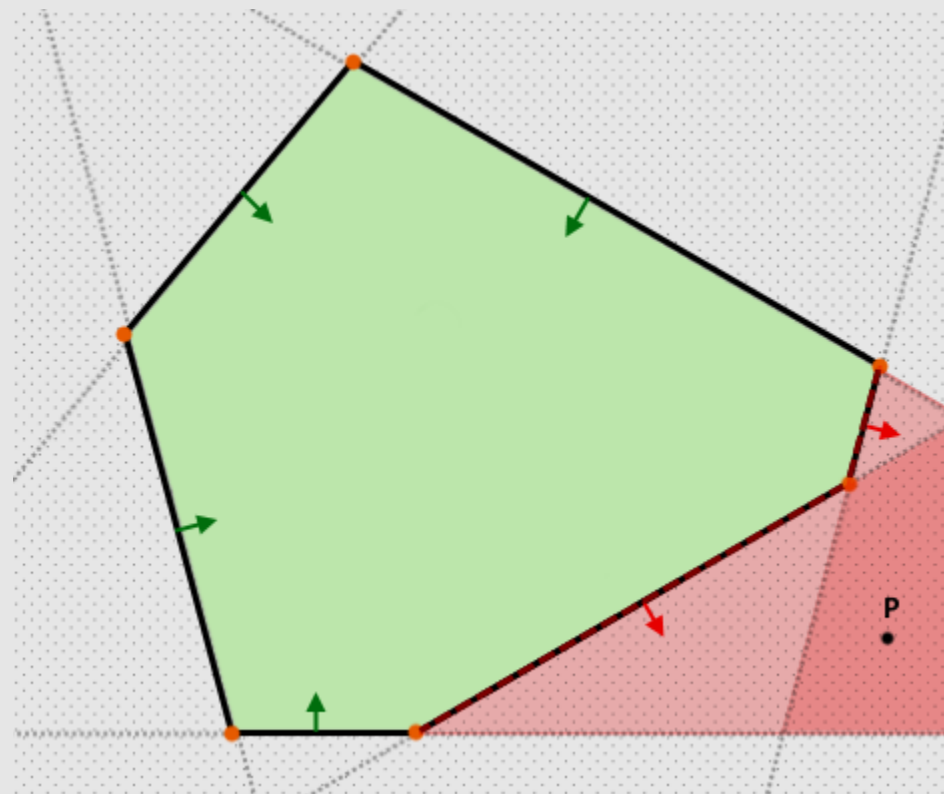
Seja o polígono convexo C e o ponto P , temos que, para P estar dentro de C é necessário que P esteja no mesmo lado em relação a todas as arestas de C .

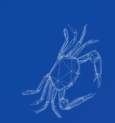




Base

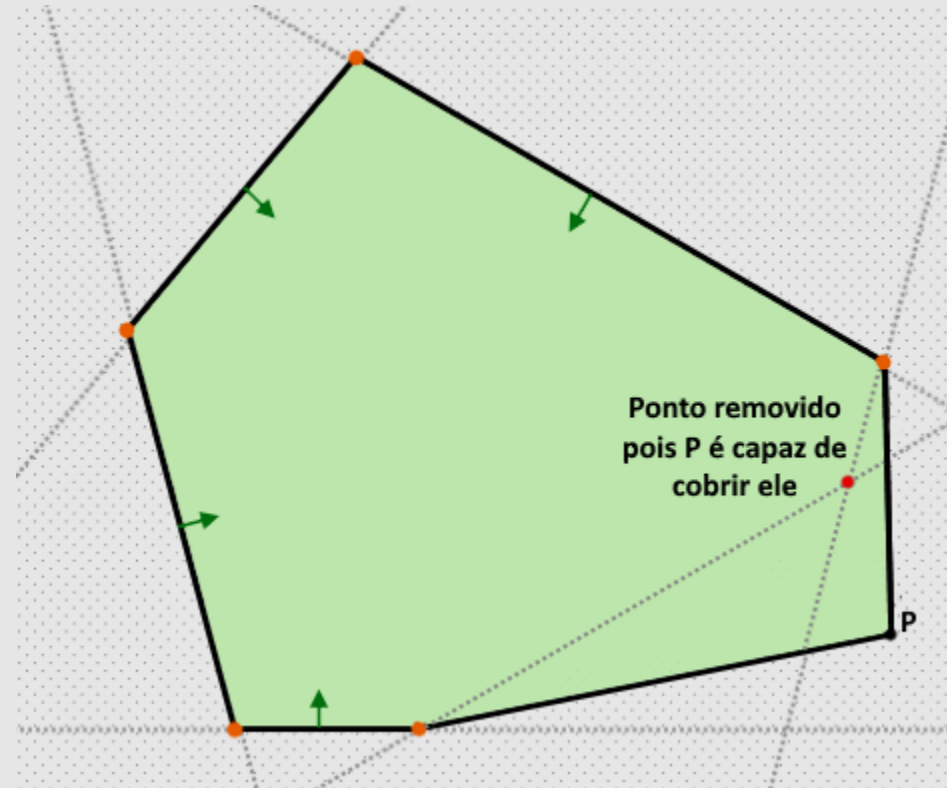
Mas caso **P** esteja fora, então teremos que **P** estará no lado errado de ao menos uma de arestas de **C**.

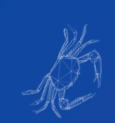




Base

Então para manter $C+P$ convexo basta ligar as extremidades das arestas válidas de C com P , gerando assim um novo fecho convexo que mantém a propriedade convexa com P .





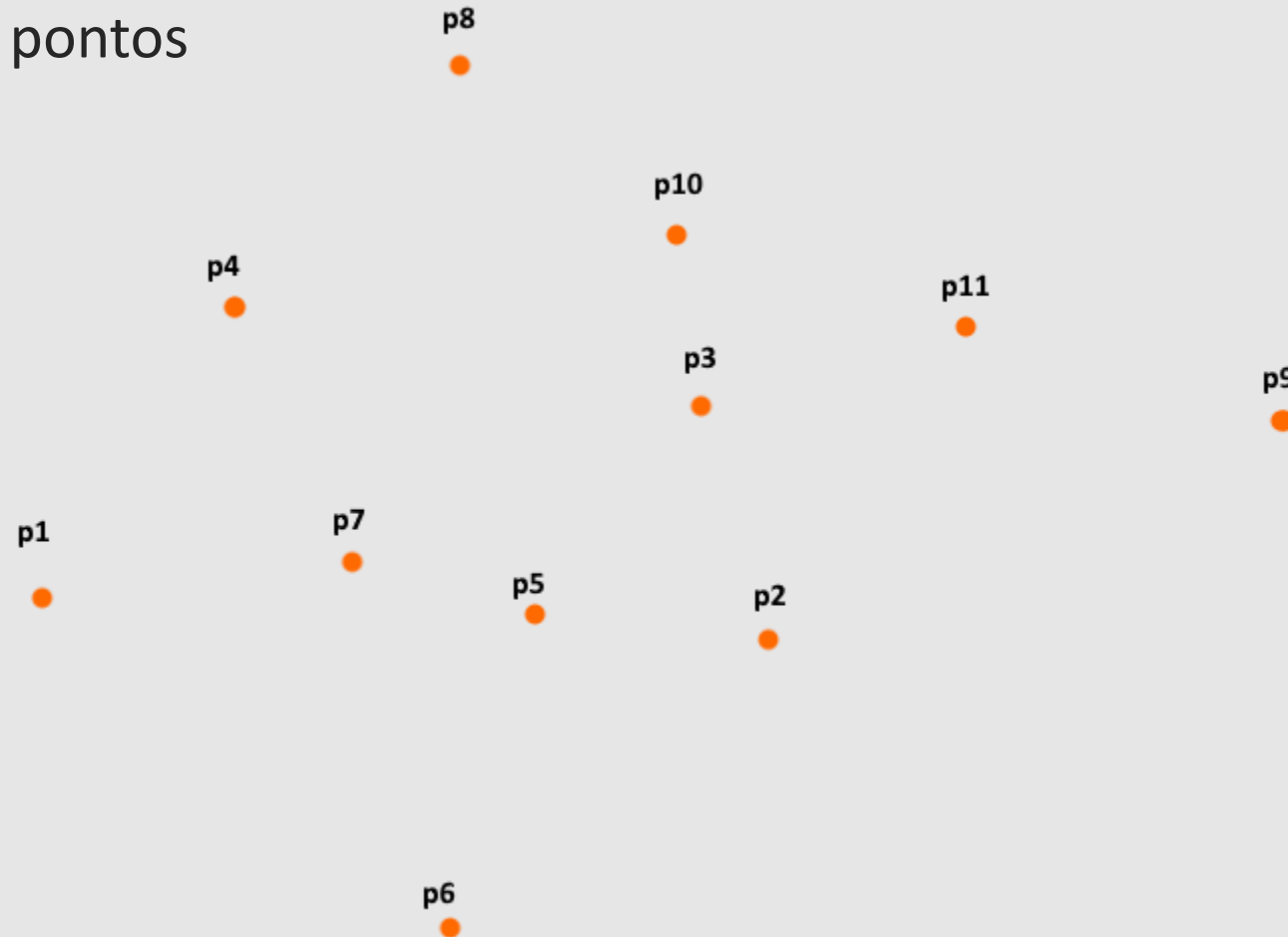
Pseudocódigo

```
Fecho_Convexo(pontos)
    fecho = Simplex(pontos);
    para cada ponto P restante:
        para cada aresta A de fecho:
            Se P está à direita de A:
                Remove_A_do_fecho();
            Liga_extremos_do_fecho_com(P);
    retorna fecho;
```



Executando o Algoritmo

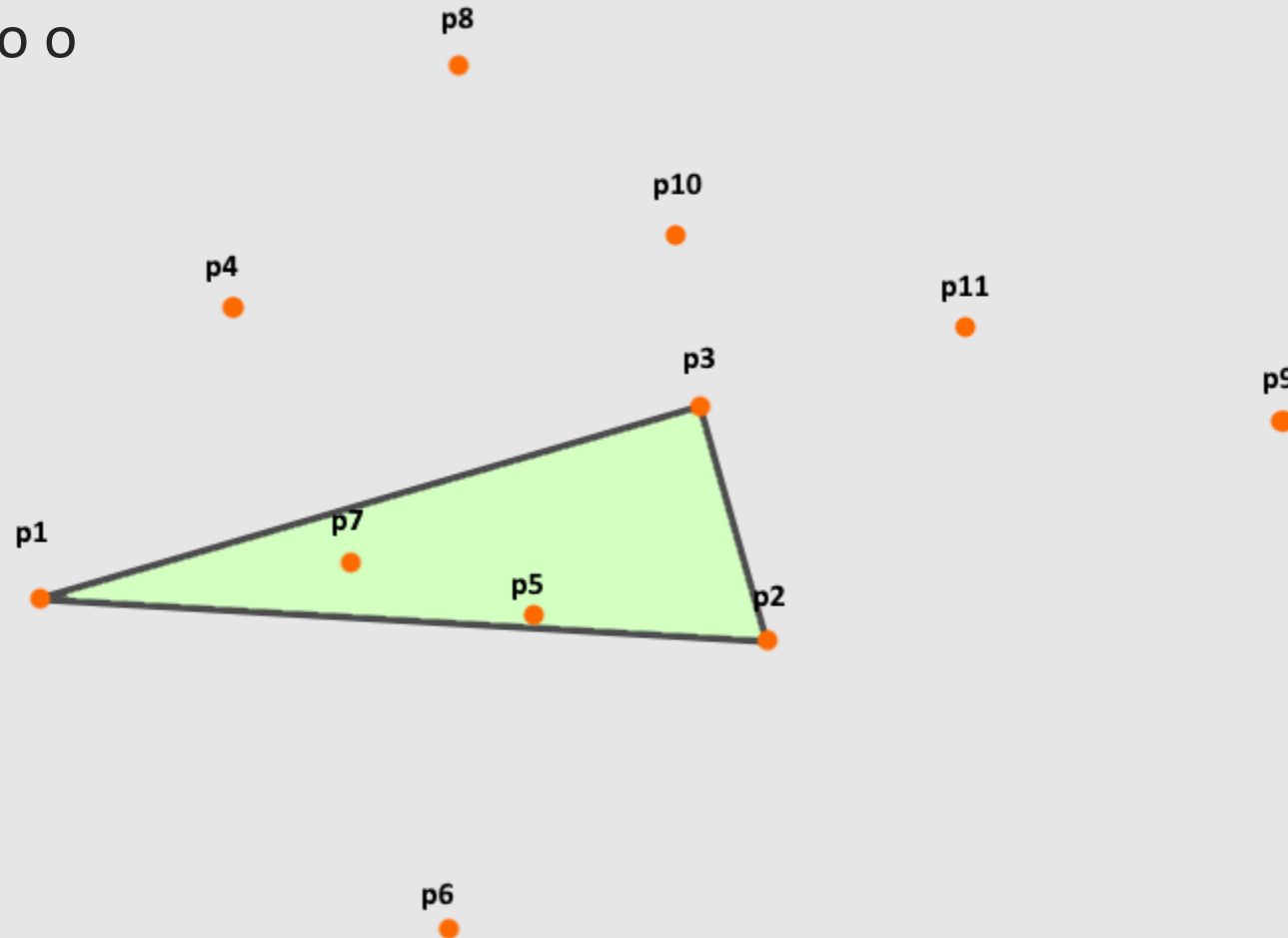
Dado o conjunto de pontos

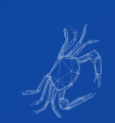




Executando o Algoritmo

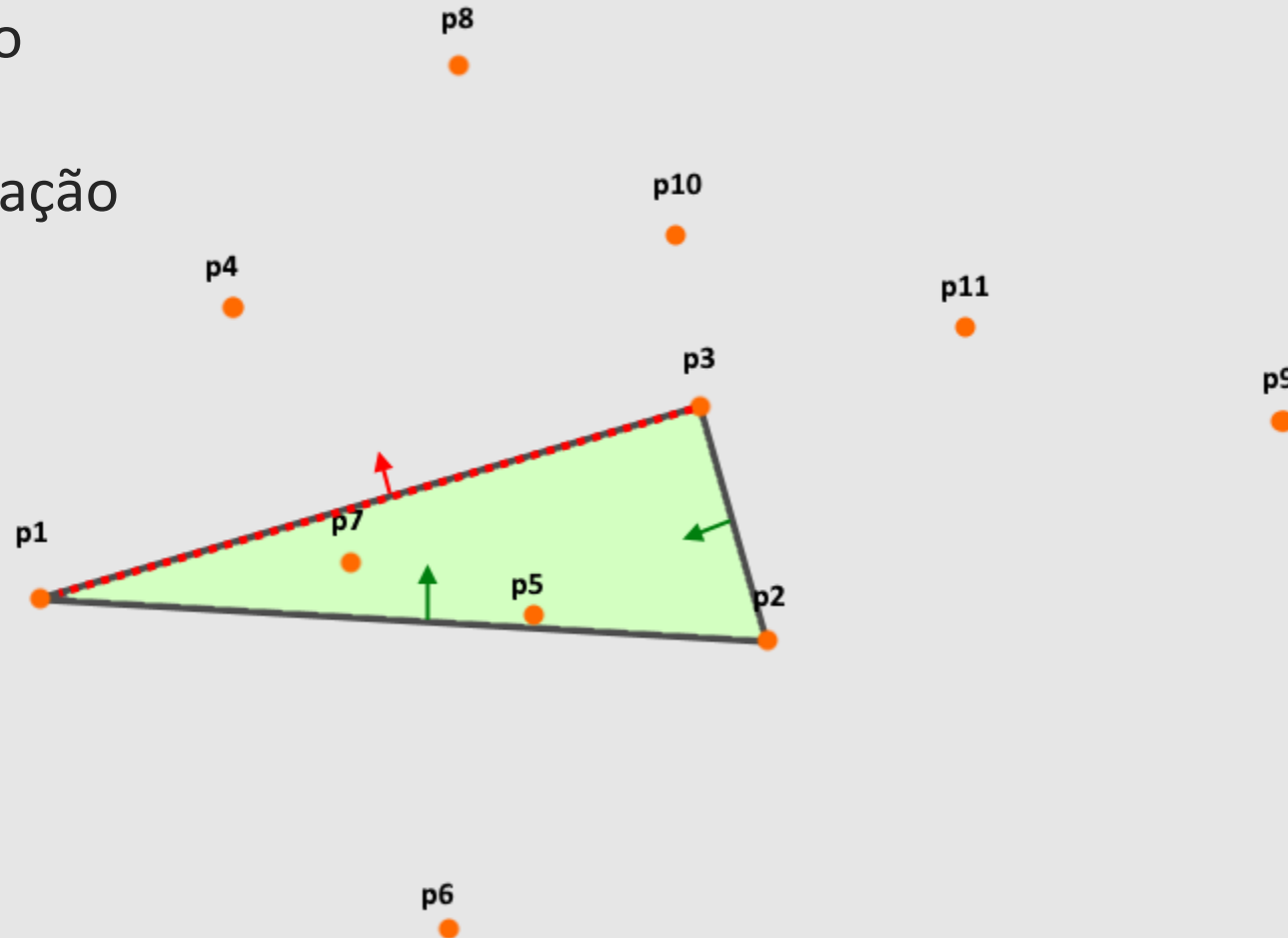
Começamos gerando o
simplex

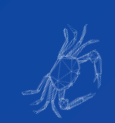




Executando o Algoritmo

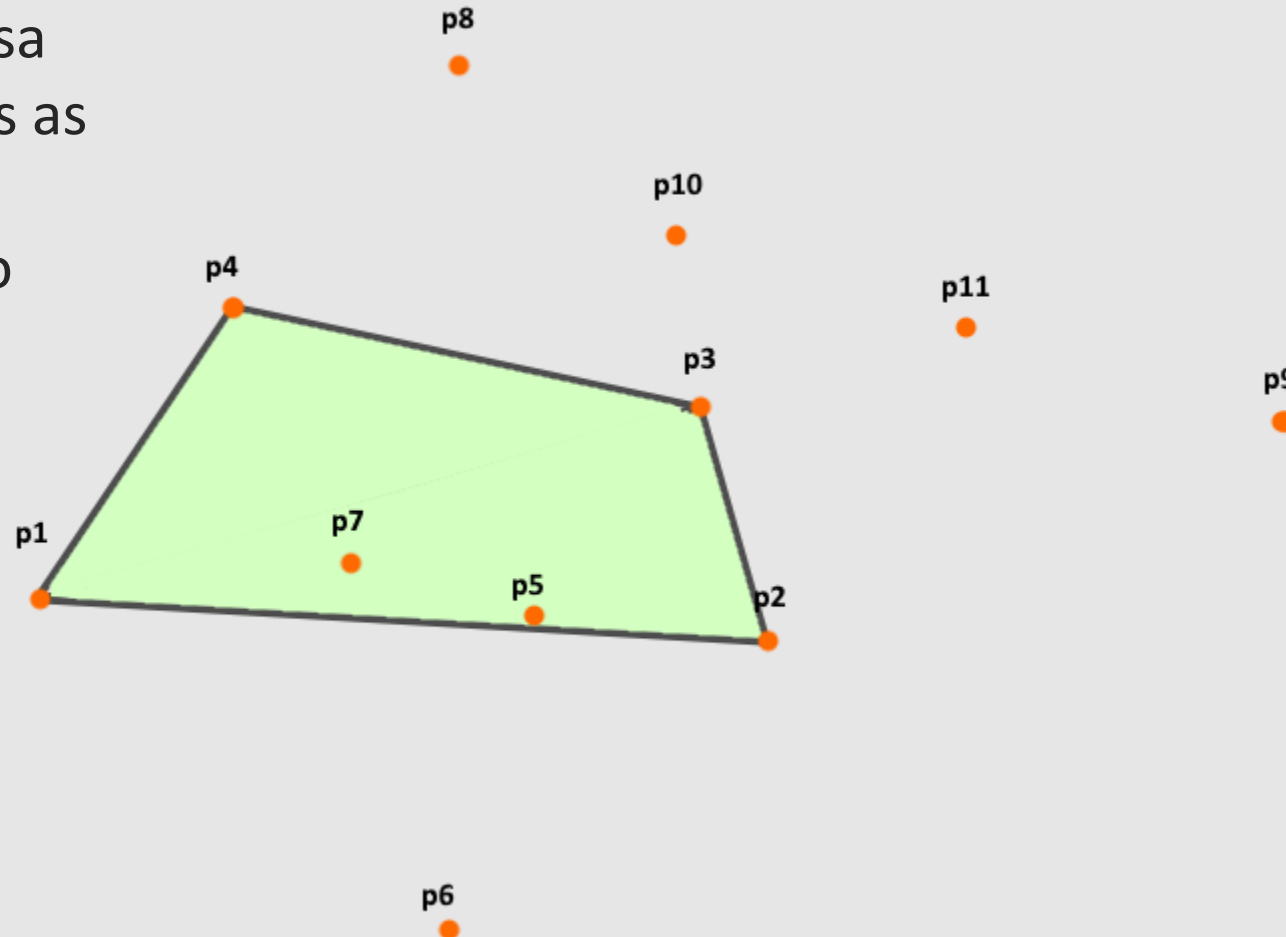
Depois vamos para o próximo ponto e verificamos a orientação dele em relação as arestas, no caso p4 está do lado Errado de uma das arestas.

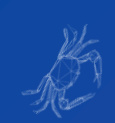




Executando o Algoritmo

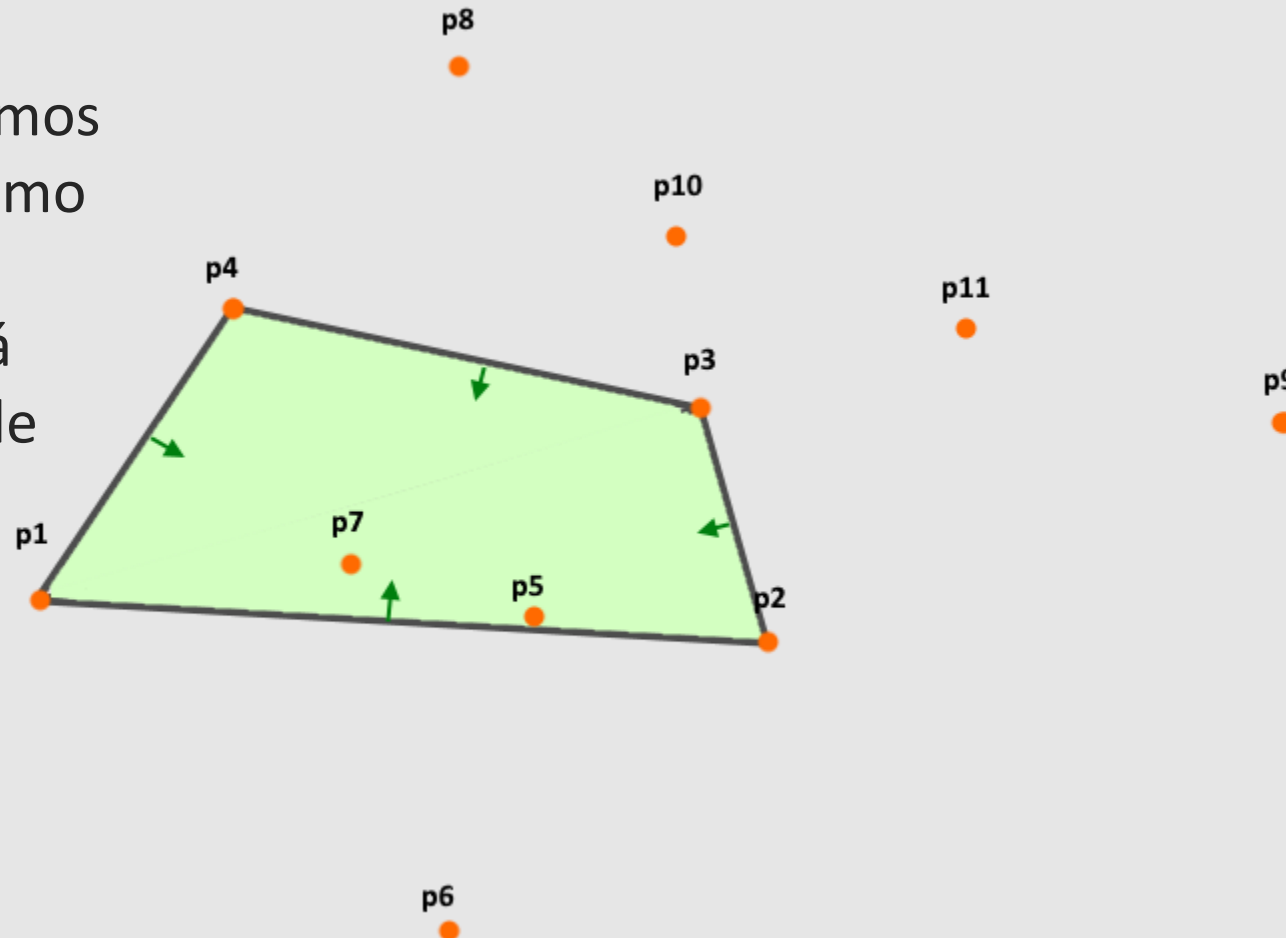
Então deletamos essa aresta e conectamos as extremidades do polígono anterior ao ponto p4





Executando o Algoritmo

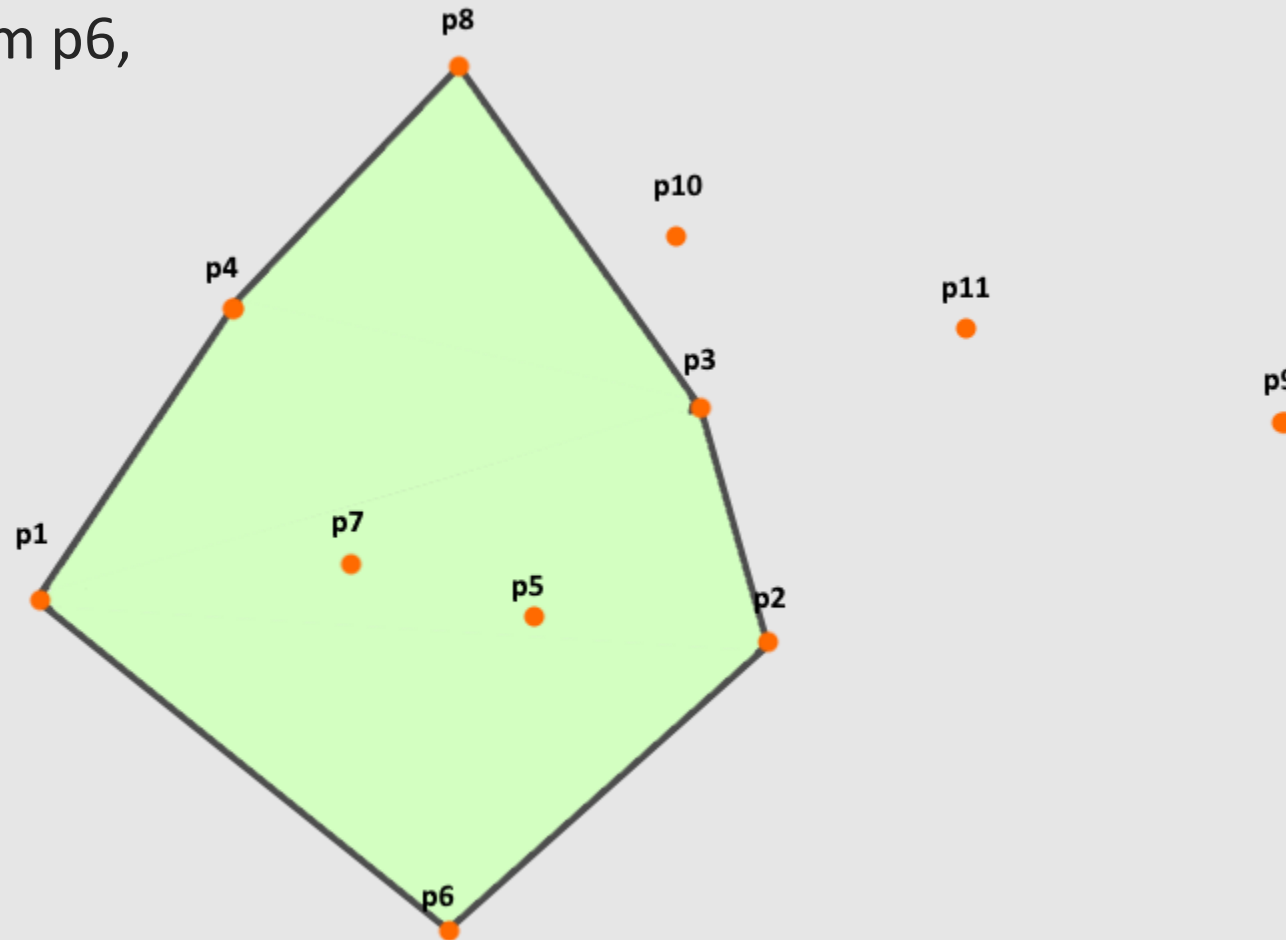
Agora vamos para o ponto p5 e percebemos que ele está do mesmo lado para todas as arestas, logo p5 está dentro do fecho e ele não precisa ser mudado





Executando o Algoritmo

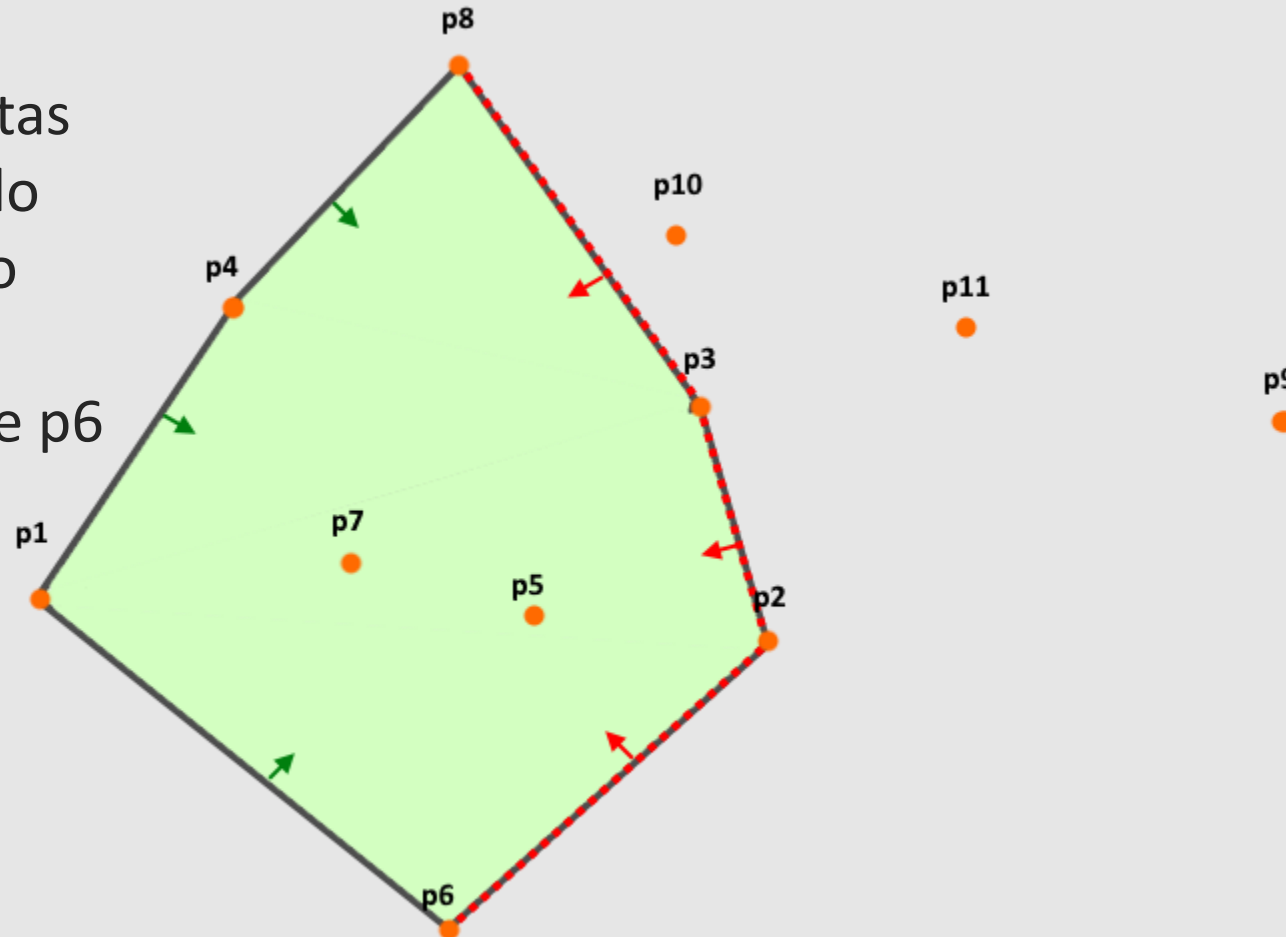
Similar acontece com p6,
p7 e p8





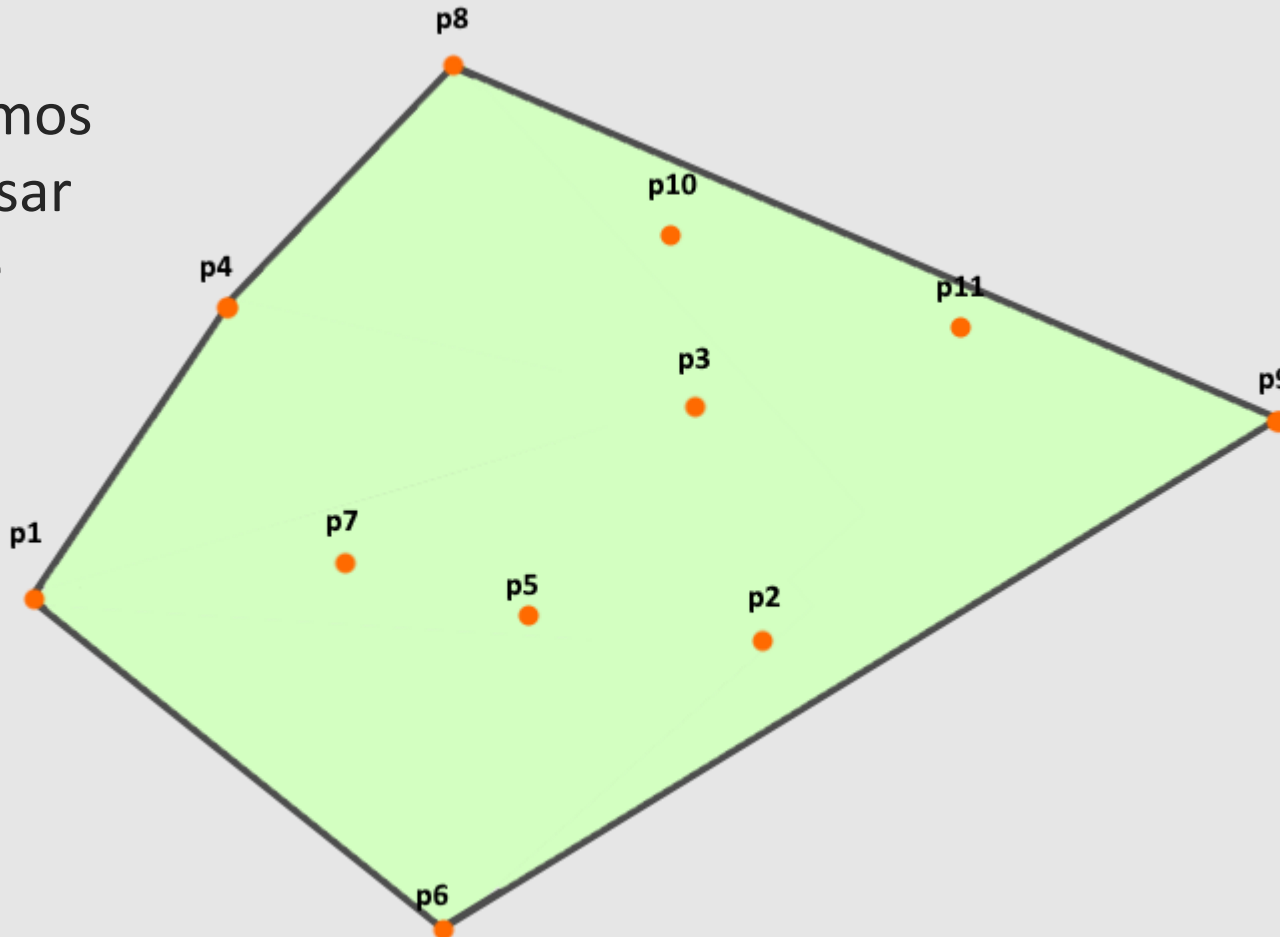
Executando o Algoritmo

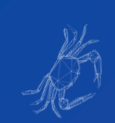
Ao analisarmos p9 encontramos 3 arestas onde ele está no lado errado, logo as 3 são deletadas deixando assim os pontos p8 e p6 como extremidades



Executando o Algoritmo

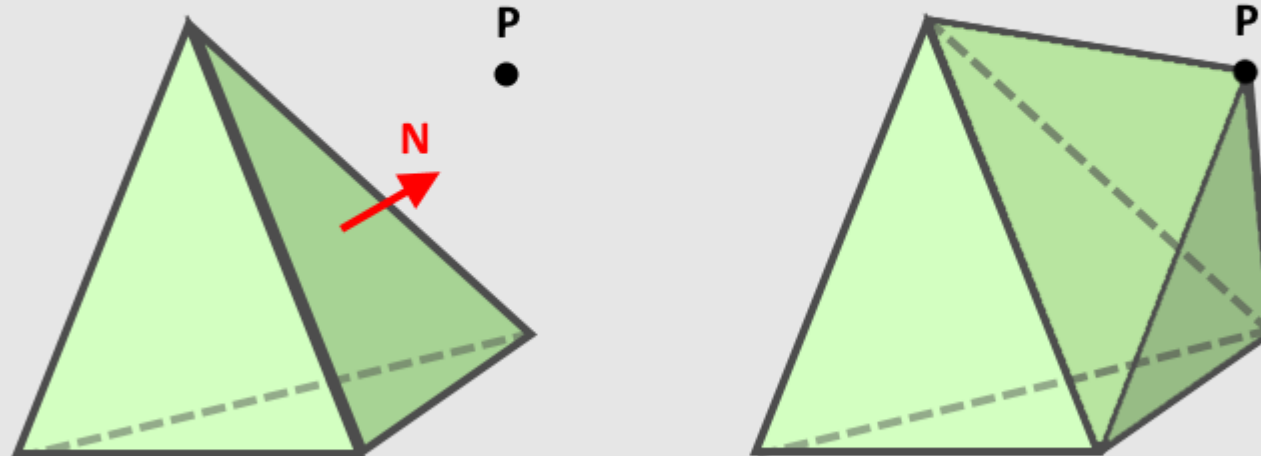
Ligamos então p9 as extremidades e obtemos o fecho, pois ao analisar p10 e p11 vemos que eles não retiram nenhuma aresta do fecho.





Para ND

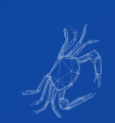
Começamos com o simplex dessa dimensão e vamos verificando em qual subespaço dos hiperplanos o ponto P se encontra, caso seja em um subespaço diferente dos demais o hiperplano é removido do fecho e um novo é gerado considerando o ponto P .





Complexidade

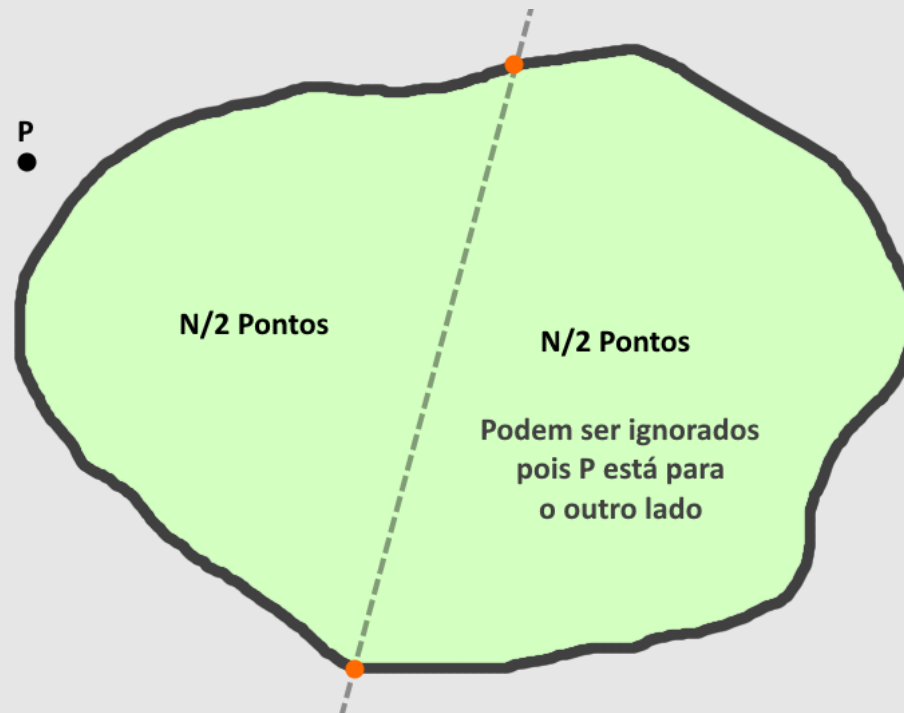
Verificamos os N pontos para ver se estão dentro ou fora das M arestas do fecho, logo a complexidade é $N \cdot M$ ou N^2 no pior caso onde todos os pontos estão no fecho e $M=N$.

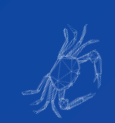


Otimizações

Otimização Inteligente:

Utilizar estruturas de busca dimensionais para não precisar comparar **P** com as **M** arestas, como por exemplo gerando uma árvore de busca baseada em coordenadas polares, tornando assim o algoritmo $n \cdot \log(m)$.

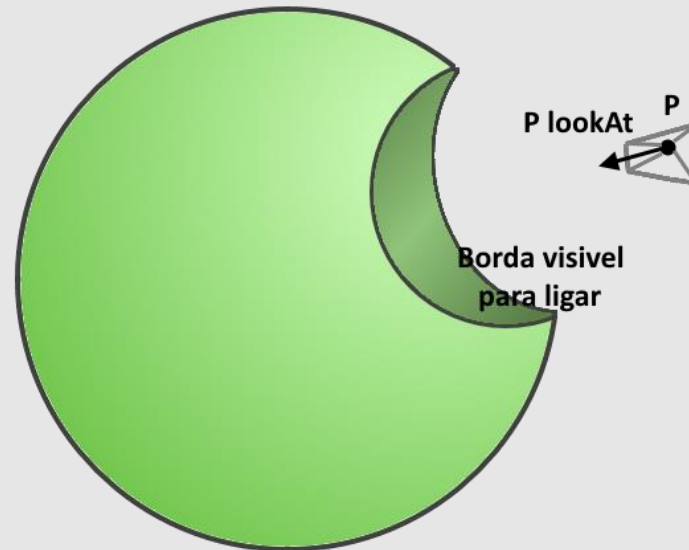




Otimizações

Otimização força bruta (GPU):

Transforma o ponto em uma câmera, uso a GPU pra rasterizar o fecho (sendo que ela é otimizada para isso) com o fecho encontro as bordas das faces visíveis, as faces visíveis serão deletadas e as bordas serão ligadas ao ponto **P** para gerar o novo fecho.





Questões?





Convex Hull

Obrigado!

