

Algoritmo para gerar invólucros convexos

Artur Queiroz - PG38014
Luís Albuquerque - PG38015

November 26, 2019

1 Introdução

Neste trabalho abordamos um dos principais temas de Geometria Computacional, Invólucros Convexos. Estes são muito usados porque ...

Existem várias formas de construir um, mas neste trabalho vamos nos cingir a implementar o "merge-hull" Que é um algoritmo que se usa, umas das técnicas mais importantes na computação, que se chama "Dividir para conquistar". Que se baseia em dividir um problema complexo, em problemas mais pequenos e mais fáceis.

2 Descrição

Para implementarmos o algoritmo apenas temos que seguir os passos a baixo.

- Ordenar os pontos por ordem lexicográfica.
- Separar os pontos em dois conjuntos A e B, onde A contem os pontos da esquerda e B os da direita.
- Calcular o invólucro convexo de A, $\mathcal{A} = I(A)$ e o de B, $\mathcal{B} = I(B)$ recursivamente
- No final juntar \mathcal{A} e \mathcal{B} , calculando o invólucro convexo de $A \cup B$.

Agora vamos explicar com mais detalhe todo o processo.

2.1 Ordenar os Pontos

Para ordenar os pontos, pode ser usado qualquer algoritmo de ordenação, tendo em atenção que a escolha do algoritmo de ordenação, pode alterar a complexidade do algoritmo como um todo. Nós optamos por escolher o algoritmo de ordenação *merge sort*, além de ter uma das melhores complexidades $\Theta(n \log n)$, achamos que se enquadra perfeitamente no espírito do algoritmo, "Dividir para conquistar".

2.2 Descrição de MergeSort

Input: array, índice esquerdo, índice direito

Começando com o índice esquerdo a 0, e a índice direito a (*tamanho do array*) - 1

- Primeiro encontra-se o índice médio do Array e divide-se em dois. ($\text{meio} = (\text{esquerda} + \text{direita})/2$)
- Calcular o MergeSort(array, esquerda, meio), com a lista que fica à esquerda
- Calcular o MergeSort(array, meio+1, direita), com a lista que fica à direita
- No final junta os dois de forma ordenada.

2.3 Separar os Pontos em dois conjuntos

2.4 Calcular o invólucro convexo

2.5 Juntar os invólucros convexos

3 Correção

Depois de mostrarmos como é o algoritmo, aqui vamos provar, porque é que o algoritmo faz o que diz que faz.

Qed.

4 Complexidade

A nossa implementação não foi exatamente igual ao algoritmo original, apesar de não alterar na conta da complexidade assintoticamente. Por isso vamos avaliar a correção da nossa implementação, e quando achamos pertinente, vamos fazer a ressalva, mencionando as diferenças em relação ao algoritmo original.

- Ordenar os pontos pela cordenada x, tem Complexidade $\Theta(n \log n)$
- Separar os pontos em dois conjuntos A e B, onde A contém os pontos da esquerda e B os da direita.
- Calcular o invólucro convexo de A, $\mathcal{A} = I(A)$ e o de B, $\mathcal{B} = I(B)$ recursivamente
- No final juntar \mathcal{A} e \mathcal{B} , calculando o invólucro convexo de $A \cup B$.

5 Conclusão