1ª Lista de Exercícios

MO619/MC948 — Geometria Computacional

Prof. Pedro J. de Rezende

1º Semestre de 2018

Comentários

- Esta lista inclui exercícios de diversas fontes.
- Pode haver exercícios parecidos ou formulações alternativas de uma mesma questão.
- Só procure resolver os exercícios **depois** de estudar a teoria e os exemplos dos livros.
- Se você estiver com dificuldades mesmo nos exercícios mais fáceis, releia a teoria mais uma vez antes de prosseguir.
- Recomendo que sejam formadas duplas de estudantes para discutir os exercícios, mas cada estudante deve tentar resolver cada exercício individualmente antes de discutir com os colegas.
- Pedir simplesmente para ver a solução de um exercício feito por um colega (ou pelo professor) é a maneira mais efetiva de destruir qualquer benefício que um exercício pode lhe trazer. Mais contribui para o aprendizado um exercício atacado insistentemente do que um transformado em mero exemplo.

1. Dados três problemas P_1, P_2 e P_3 e as reduções $P_1 \propto_n P_2$ e $P_2 \propto_{n^2} P_3$, diga quais das afirmações abaixo são falsas e justifique.

   (a) Se P_2 possui cota inferior $\Omega(h(n))$, então P_3 também possui cota inferior $\Omega(h(n))$.

   (b) Se P_2 possui cota superior $O(h(n))$, então P_1 possui cota superior $O(h(n) + f(n))$.

   (c) Se P_2 for um problema de enumeração, então P_1 é um problema de decisão e P_3 pode ser qualquer tipo de problema (decisão, contagem ou enumeração).

   (d) Se P_3 for um problema de decisão, então P_1 e P_2 também devem ser problemas de decisão.

2. Dada uma coleção de n círculos definidos pelos seus centros e raios prove que o problema de decidir se existe alguma interseção entre quaisquer dois desses círculos possui cota inferior $\Omega(n \log n)$ no modelo de árvore de decisões algébricas.

3. Dado um conjunto S de n pontos em $\mathbb{R}^2$, descreva um algoritmo que, após pré-processamento, determina, em tempo $O(\log n)$, se um ponto qualquer $p \in \mathbb{R}^2$ dado é uma combinação linear convexa de algum trio de pontos de S. Qual o tempo gasto na etapa do pré-processamento e qual o espaço requerido?

4. Projete um algoritmo com tempo de execução $O(n)$ que ao receber um vértice v_n e um polígono P determina se P é estrelado em relação a v_n.

5. O retângulo envolvente de área mínima de um conjunto S de pontos no plano, definido como o menor retângulo que contém no seu interior todos os pontos de S é útil para agilizar a detecção de colisão entre objetos em jogos eletrônicos.

   (a) É verdade que um dos lados desse retângulo sempre contém um dos lados do casco convexo dos pontos de S? Justifique.

   (b) Descreva um algoritmo eficiente que determine os vértices do retângulo envolvente de área mínima de um dado conjunto de n pontos. Analise a complexidade.

6. Sejam P e Q dois polígonos convexos com m e n vértices, respectivamente. Escreva as expressões correspondentes aos valores máximo e mínimo do número de vértices do casco convexo de $P \cup Q$, em termos de m e n, para os seguintes casos:

   - P está inteiramente contido no interior de Q;
   - $P - Q \neq \emptyset$ e $Q - P \neq \emptyset$;
   - P e Q são disjuntos.

7. Considere o polígono estrelado definido por $P = \{(1,1); (5,1); (5,5); (4,5); (3,4); (3,3); (4,2); (3,2); (2,3); (2,4); (3,5); (1,5)\}$. Execute passo a passo o algoritmo da varredera de *Graham*, explicitando cada passo na solução desta questão.

8. Por que podemos dizer que o algoritmo de Andrew para o casco convexo é uma especialização do algoritmo de varredura de Graham para o casco convexo?

9. O Algoritmo 4.4 do livro [dRS94] §4.3.4 constrói um polígono estrelado a partir de um conjunto de pontos, dado um vértice inicial. Dependendo da escolha do vértice inicial, o polígono estrelado gerado pode ser diferente. Mostre que, se para qualquer vértice inicial escolhido o polígono estrelado gerado é sempre o mesmo, então estes polígonos são convexos. (Obs: dois polígonos são considerados iguais se suas listas *circulares* de vértices forem iguais.)

10. Descreva um algoritmo linear para encontrar o casco convexo da união de dois polígonos convexos dados.

11. Dado um conjunto de pontos S, descreva um método que determine se um ponto $p \in S$ pertence ao casco convexo de S em tempo $O(n)$.

12. Dado um conjunto de pontos, a profundidade de cada ponto é definida como o número da camada das cascas convexas do conjunto da qual o ponto é vértice.

    Prove o Teorema 4.13 de [PS85]:

    "Qualquer algoritmo que determina a profundidade de cada ponto em um conjunto tem cota inferior de $\Omega(n \log n)$, onde n é o número de pontos."

13. Apresente um algoritmo incremental para o problema de se encontrar o casco convexo de um conjunto de pontos. Que modificações seriam necessáriam para transformar esse algoritmo em uma abordagem de varredura? Faça a análise de complexidade dos algoritmos.

14. Faça uma prova por indução da qual se pode extrair o algoritmo por divisão e conquista para a construção do casco convexo.

15. São dados n pontos distintos no plano. Suponha que a envoltória convexa já é conhecida. Considere a remoção de um ponto que é vértice da envoltória. Forneça um algoritmo que encontra a nova envoltória convexa. É possível utilizar a envoltória que já tinha sido calculada? Qual a complexidade do seu algoritmo?

16. Seja $K \in \mathbb{Z}$ uma constante. Considere um reticulado ortogonal $K \times K$. Dado um conjunto S de $n < K^2$ pontos sobre o reticulado, projete um algoritmo para construção do casco convexo de S em tempo $O(n)$. Quais operações deve ter um modelo computacional em que seu algoritmo seja descritível.

17. Diz-se que um ponto $p = (x_p, y_p)$ *domina* um outro ponto distinto $q = (x_q, y_q)$ se $x_p \geq x_q$ e $y_p \geq y_q$. Um ponto p é dito *maximal* de um conjunto S se $p \in S$ e nenhum ponto de S domina p.

    Descreva um algoritmo que em tempo $O(n \log n)$ determina todos os pontos maximais de um conjunto de n pontos distintos no plano. (Sugestão: use varredura planar.)

18. (a) Dados dois polígonos convexos P e Q, mostre como encontrar o polígono intersecção de P e Q. Qual a complexidade de seu algoritmo?

    (b) O polígono intersecção é sempre convexo para quaisquer polígonos convexos P e Q? Justifique.

19. Considere o seguinte problema.
    Dado um conjunto S de pontos no plano, pré-processá-los de modo a poder responder rapidamente a consultas do seguinte tipo: dados dois valores não negativos a e b,

determinar qual o maior número de pontos de S que podem estar dentro de um retângulo de lados a e b.

(a) Modifique um algoritmo de busca por amplitude de forma a resolver esse problema eficientemente.

(b) Analise o algoritmo do item anterior.

20. Considere o problema de localização de pontos em relação a um dado polígono simples.

    (a) Descreva um algoritmo que, após pré-processamento do retângulo dado, realiza a consulta de cada ponto em $O(\log n)$

    (b) Quais são as complexidades deste algoritmo em relação ao pré-processamento e ao espaço de armazenamento exigido, no pior caso?

21. Mostre que uma árvore 2D como a construída para o problema de busca por amplitude possui altura $\log n$. Qual a característica que garante essa propriedade?

22. Descreva um algoritmo $O(\log n)$ para determinar a distância entre um ponto no plano e um polígono convexo de n lados.

23. Dados $4n$ pontos no plano, metade dos quais são brancos e metade são azuis, descreva um algoritmo que encontra uma reta que deixa n pontos azuis e n pontos brancos de cada lado. Em que modelo computacional seu algoritmo pode ser descrito? Qual a complexidade de seu algoritmo?

24. Dados dois conjuntos de pontos A e B, modifique o algoritmo que encontra a menor distância entre dois pontos visto em classe, para que calcule a menor distância entre 2 pontos p e q, com $p \in A$ e $q \in B$.

25. Seja P um problema que tem como entrada um conjunto de pontos S e um número d e deseja-se determinar se existe um par de pontos em S que possui distância exatamente d. Comparando P com o problema do par mais próximo, P parece mais simples, visto que deseja-se apenas <u>localizar</u> um par com distância d e não determinar o par que tem <u>a menor</u> distância. Sendo assim, é possível modificar o algoritmo do par mais próximo de modo que a localização de um par de pontos com distância d seja feita em tempo menor que a determinação do par mais próximo? Explique.

26. A fim de resolver o problema do Par Mais Próximo em duas dimensões (PMP2D), um aluno da disciplina de MO619 desenvolveu um algoritmo de varredura planar. Seja $S = \{p_1, \ldots, p_n\}$ o conjunto de pontos ordenados pelas abcissas obtidos a partir dos pontos dados como entrada, temos que o algoritmo baseia-se na seguinte invariante: a cada vez que um novo ponto p_i é visitado pela linha de varredura (i-ésimo evento),

a menor distância δ entre os pontos $\{p_1, \ldots, p_{i-1}\}$ é conhecida. Assim, para cada evento, o algoritmo calcula as distâncias entre o ponto corrente e os demais pontos já visitados pela linha varredura, atualizando δ caso alguma das distâncias calculadas seja menor do que δ.

(a) O algoritmo citado anteriormente gasta tempo $O(n)$ cada vez que um evento é processado. Mostre que é possível realizar tal processamento de maneira mais eficiente.

(b) Escreva o pseudo-código e faça análise de complexidade de um algoritmo de varredura que resolve PMP2D em tempo $O(n \log n)$.

(c) Demonstre que a invariante citada no enunciado é verdadeira para o seu algoritmo, provando assim sua corretude.

27. Qual a complexidade de pré-processamento, armazenagem e consulta do método das fatias e do método das cadeias para busca geométrica em subdivisão planar. Escolha um dos métodos e justifique sua resposta para complexidade de consulta.

28. Diferencie os paradigmas incremental, de varredura e divisão e conquista usados para o projeto de algoritmos. Dê um exemplo de um problema geométrico e uma breve explicação de como cada paradigma o resolve.

29. O cientista von Genius acaba de iniciar uma pesquisa sobre algoritmos para guiar formigas robóticas. Atualmente, cada formiga i se movimenta no plano segundo duas funções polinomiais, parametrizadas pelo tempo, expressas por $(x_i(t), y_i(t))$ com $i = 1, 2, \ldots, n$ e $t \in \mathbb{Z}$. O custo de montagem de uma só formiga chega a ser centenas de reais e por isso uma câmera as monitora com o intuito de evitar colisões. Para tanto, o sistema aciona um mecanismo que força uma parada em todas as criaturas, caso seja detectado que pelo menos duas delas irão colidir no instante $t + 1$.

    Seu desafio é elaborar um algoritmo $O(n \log n)$ que detecta se irá ocorrer colisão no instante $t + 1$, tendo como entrada os polinômios e o valor atual de t. (Uma colisão ocorre quando duas formigas se encontram em uma mesma posição.)

30. Seja C um circunferência sobre a qual são colocados n pontos. Conecte cada ponto com cada um dos outros, com segmentos de reta. Desta forma, tem-se uma subdivisão do interior da circunferência em faces. Assumindo que por cada ponto de intersecção desses segmentos (no interior da circunferência) passam exatamente dois (e nunca três!) segmentos, calcule o número de regiões formadas e prove a sua resposta.