CPS740 - Prova 1

Pedro Maciel Xavier
116023847

23 de agosto de 2020

Questão 1.:

1) Se G é 3-colorível, então G possui um ciclo ímpar. **Falso.**

Um grafo G k-colorível é aquele para o qual existe uma k-coloração, isto é, uma função $\mathcal{C} : V(G) \to [r]$[1] tal que $(v, w) \in E(G) \implies \mathcal{C}(v) \neq \mathcal{C}(w)$. Um grafo é k-colorível tem seu número cromático $\chi(G) \leq k$.

Como dito no livro-texto, colorir um grafo de n vértices com n cores é uma tarefa muito simples, basta colorir cada vértice com uma cor distinta. Assim, podemos dizer que todo grafo de n vértices é n-colorível. Para construir um contraexemplo ao enunciado, basta colorir um grafo de 3 vértices que não possua um ciclo como o do K_3.

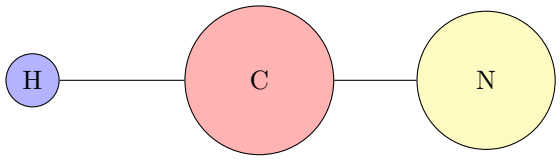


Figura 1: 3-coloração sem ciclos sobre o Cianeto de Hidrogênio (CHN).

2) Seja $G(V_1 \cup V_2, E)$ um grafo bipartido conexo. Então o grafo complementar G^c também é conexo. **Falso.**

Como V_1 e V_2 são conjuntos independentes, é evidente que o grafo complementar contará com duas cliques, formadas pelos vértices de V_1 e V_2, respectivamente.

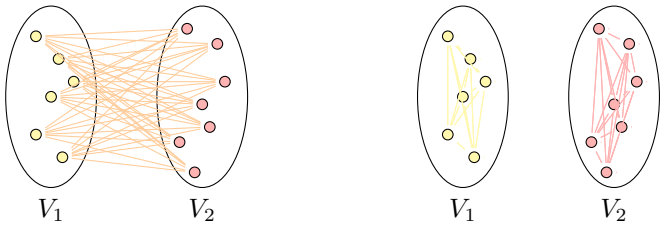


Figura 2: O Grafo $G(V_1 \cup V_2, E) = K_{|V_1|,|V_2|}$ e seu complemento G^c

No entanto, no caso do grafo bipartido completo $K_{|V_1|,|V_2|}$ temos que seu complemento não possui nenhuma aresta que ligue um vértice de V_1 a algum outro em V_2. Temos assim um contraexemplo.

3) Se retirarmos uma aresta qualquer do grafo $K_{3,3}$, o grafo resultante é planar. **Verdadeiro.**

O **Teorema 2.7**[1] afirma que um grafo é planar se e somente se não possuir nenhum subgrafo que seja uma subdivisão de K_5 ou $K_{3,3}$. A subdivisão de um grafo G consiste em inserir um vértice $u \notin V(G)$ ao grafo após a remoção de uma aresta $(v, w) \in E(G)$ seguida da adição das arestas (v, u) e (u, w).

[1]Notação para o conjunto dos números naturais até n, $[n] = \{1, 2, \ldots, n\}$

Seja $\pi_{v,w}(G)$ uma subdivisão do grafo G através da aresta (v, w). Sabemos, pela construção deste processo, que $|V(\pi_{v,w}(G))| = |V(G)| + 1$ e que $|E(\pi_{v,w}(G))| = |E(G)| + 1$. Conclui-se que, sendo $\pi(G)$ uma subdivisão qualquer de G, temos que $|E(\pi(G))| \geq |E(G)|$. Seja G o grafo obtido removendo uma aresta de $K_{3,3}$, temos que todo subgrafo H de G possui $|E(H)| \leq |E(G)|$. Como $|H(G)| \leq |E(G)| < |E(K_{3,3})|$, é claro que nenhum subgrafo H de G é subdivisão de $K_{3,3}$.

Da mesma forma, nenhum subgrafo próprio de $K_{3,3}$ é capaz de conter uma subdivisão de K_5, já que pra isso é necessário possuir ao menos 5 vértices de grau 5. Portanto, qualquer grafo $G = K_{3,3} - (v, w)$ tal que $(v, w) \in E(K_{3,3})$ é planar.

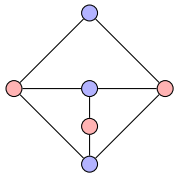


Figura 3: Um grafo bipartido planar

Outro raciocínio possível seria verificar pela remoção de uma das arestas que se obtêm um grafo planar. Em seguida, basta observar que todos os grafos obtidos pela remoção de uma aresta de $K_{3,3}$ são isomorfos e, portanto, planares.

4) Seja f um isomorfismo de um grafo G para um grafo H, e seja w um vértice em G. O grau de w em G é igual ao grau de $f(w)$ em H. **Verdadeiro.**

Dados dois grafos, G e H, dizemos que $G \cong H$ se $\exists f : V(G) \to V(H)$ tal que

$$(w, v) \in E(G) \iff (f(w), f(v)) \in E(H) \tag{1}$$

Satisfeito, f é dito um isomorfismo entre G e H. Seja $w \in V(G)$ um vértice qualquer onde $f(w) \in V(H)$,

Suposição. $grau(w) = grau(f(w))$

Demonstração.

$$\text{Seja } \mathbb{I}_{\Omega}\{\omega\} \triangleq \begin{cases} 1 \text{ se } \omega \in \Omega \\ 0 \text{ caso contrário} \end{cases}$$

$$\begin{aligned} \text{grau}(w) &= \sum_{v \in V(G)} \mathbb{I}_{E(G)}\{(w, v)\} && (2) \\ &= \sum_{f(v) \in V(H)} \mathbb{I}_{E(H)}\{(f(w), f(v))\} && (3) \\ &= \text{grau}(f(w)) \end{aligned}$$

■

De (2) para (3) utilizamos a relação (1), extraída da definição de isomorfismo em grafos presente no Capítulo 2 do livro[1]. Utilizamos também o

fato de que $V(H) \cong f(v) : v \in V(G)$, da definição de domínio do isomorfismo.

5) O Grafo abaixo é planar: **Falso.**

Seja $\pi(G)$ uma subdivisão qualquer de G. Invocando mais uma vez o **Teorema 2.7**[1], vamos buscar por subdivisões de K_5 e $K_{3,3}$ no grafo. Certamente não há nenhuma instância de $\pi(K_5)$, visto que só existe um vértice que possui grau maior ou igual a 5, quando são necessários ao menos 5.

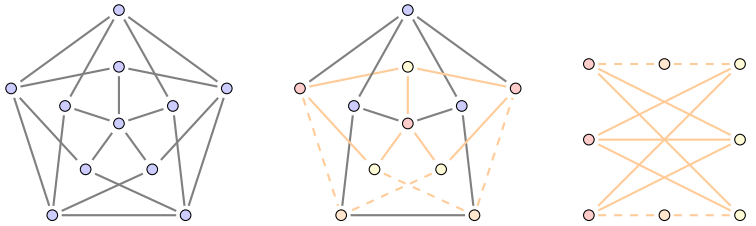


Figura 4: Grafo que não é planar pois esconde em si o $K_{3,3}$.

No entanto, encontramos uma subdivisão de $K_{3,3}$ e podemos afirmar que o grafo não é planar.

6) Todo hipercubo de dimensão n, $n \geq 1$, possui ciclo Hamiltoniano. **Falso.**

Seja $\mathscr{H}_n$ o n-ésimo hipercubo, um grafo com 2^n vértices de grau n.

Construímos $\mathscr{H}_k$ utilizando duas cópias de $\mathscr{H}_{k-1}$, notadamente $\mathscr{H}_{k-1}$ e $\mathscr{H}'_{k-1}$, ligando os vértices das duas componentes através da adição de arestas $(w, f(w)), w \in \mathscr{H}_{k-1}, f(w) \in \mathscr{H}'_{k-1}$ definidas pelo isomorfismo trivial $f : \mathscr{H}_{k-1} \to \mathscr{H}'_{k-1}$. O processo é ilustrado pela figura 5.

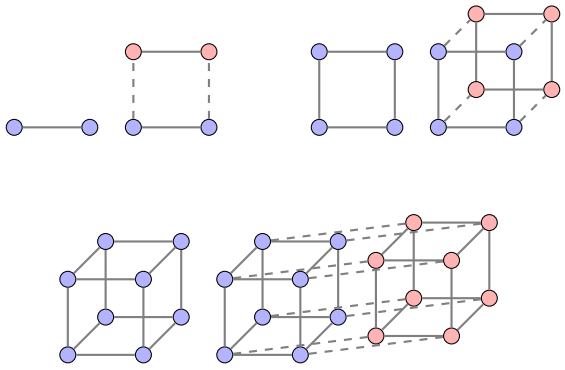


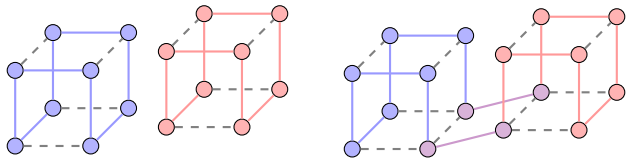
Figura 5: A construção de um Hipercubo

Assim, o método descrito acima, que constrói e define a família de grafos $\mathscr{H} = \{\mathscr{H}_k : k \in \mathbb{N}\}$, pode ser descrito de maneira aritmética e geométrica para além da intuição topológica presente na figura.

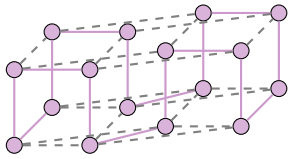
Comecemos com o conjunto de vértices $\mathscr{V}_1 = \{0, 1\}$ representados por vetores binários de tamanho 1. Para obter o conjunto $\mathscr{V}_k = V(\mathscr{H}_k)$, basta calcular $\mathscr{V}_k = \mathscr{V}_{k-1} \times \mathscr{V}_1$. O conjunto $E(\mathscr{H}_n)$ de arestas do hipercubo é definido por $\mathscr{E}_n = \{(v, w) : v, w \in \mathscr{V}_n, v \oplus w = 2^m, m \in \mathbb{N} \cup \{0\}\}$, onde

$x \oplus y$ denota a operação `xor` (ou-exclusivo) *bit*-a-*bit*. Esta representação é útil pois indica uma imersão natural dos vértices de $\mathscr{V}_n$ em $[0,1]^n \subseteq \mathbb{R}^n$.

Vamos utilizar como caso base de indução o hipercubo $\mathscr{H}_2$, que é um ciclo hamiltoniano por si só. Como passo de indução, supomos que um hipercubo $\mathscr{H}_{k-1}$ possui um ciclo hamiltoniano $\mathscr{C}_{k-1}$ do qual uma aresta (v, w) faz parte. Fazemos uma cópia $\mathscr{H}'_{k-1}$ de $\mathscr{H}_{k-1}$ para construir $\mathscr{H}_k$ segundo o processo já explicitado. Para compor o ciclo $\mathscr{C}_k$, vamos remover a aresta (v, w) de $\mathscr{C}_{k-1}$ assim como a aresta (v', w') de $\mathscr{C}'_{k-1}$ (ciclo Hamiltoniano em $\mathscr{H}'_{k-1}$). Por fim, adicionamos as arestas (v, v') e (w, w') ao subgrafo $\mathscr{C}_{k-1} \cup \mathscr{C}'_{k-1}$ para obter $\mathscr{C}_k$.



$\mathscr{C}_{k-1}$ em $\mathscr{H}_{k-1}$ Passo indutivo



$\mathscr{C}_k$ em $\mathscr{H}_k$

Figura 6: Ciclo Hamiltonianos em Hipercubos

Com este argumento, fica demonstrado que todo hipercubo $\mathscr{H}_n, n \geq 2$ possui ciclo Hamiltoniano. No entanto, isso não vale para $n = 1$, já que $\mathscr{H}_1$ possui apenas 2 vértices e não é capaz de conter ciclos.

Questão 2.:

Vamos supor que utilizamos r_1 cores em uma coloração ótima para G_1 e que precisamos de ao menos r_2 cores para colorir G_2. Seja $r = \max\{r_1, r_2\} = \max\{\chi(G_1), \chi(G_2)\}$, vamos usar no máximo r cores para pintar os dois grafos separadamente.

Sejam $C_1 : V(G_1) \to [r]^2, C_2 : V(G_2) \to [r]$ colorações para os vértices de G_1 e G_2, respectivamente. Chamaremos $G = \sigma_{v,w}(G_1, G_2)$ o grafo resultante da identificação dos vértices $v \in V(G_1)$ e $w \in V(G_2)$, escolhidos ao acaso, em um único vértice $s \in V(G)$. Diante dessas informações temos os seguintes casos possíveis:

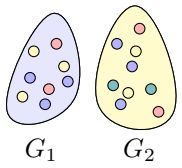


Figura 7: Coloração ótima de cada grafo.

I. Se $C_1(v) = C_2(w) = r^* \in [r]$, podemos identificar os vértices sem alterar a coloração de cada grafo. Isto é, construir a coloração $C : V(G) \to [r]$ através de C_1 e C_2:

$$C(u) \triangleq \begin{cases} C_1(u) \text{ se } u \in V(G_1) \\ C_2(u) \text{ se } u \in V(G_2) \\ r^* \text{ caso contrário} \end{cases}$$

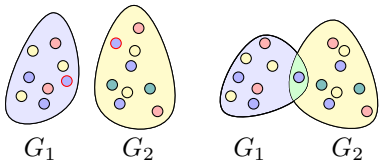


Figura 8: Identificação de vértices de mesma cor.

Dessa maneira, para colorir $G = \sigma_{v,w}(G_1, G_2,)$ é garantido que não será necessária nenhuma cor que não esteja em $[r]$, ou seja, $\chi(G) \leq r$.

Teorema. *Seja H subgrafo de G, então $\chi(H) \leq \chi(G)$*

Demonstração. Encontramos uma coloração válida para H ao supor que o grafo G possui uma coloração ótima $\mathcal{C} : V(G) \to [\chi(G)]$. Aplicamos a mesma coloração sobre H, com a garantia de que vértices adjacentes possuem cores distintas. ■

[2]Notação para o conjunto dos números naturais até n, $[n] = \{1, 2, \ldots, n\}$

Pelo teorema acima, como G_1 e G_2 são subgrafos de $G = \sigma_{v,w}(G_1, G_2)$, é evidente que teremos tanto $\chi(G) \geq \chi(G_1)$ como $\chi(G) \geq \chi(G_2)$. Portanto, $\chi(G) \geq \max\{\chi(G_1), \chi(G_2)\} = r$. Com $\chi(G) \leq r$ e $\chi(G) \geq r$ concluímos que $\chi(G) = r$.

II. Caso $C_1(v) \neq C_1(w)$, vamos construir uma coloração alternativa C_2' para G_2. Queremos que $C_2(w)' = C_1(v)$. Vamos, portanto, colorir os vértices cuja cor é a mesma de v com a cor de w, a medida que colorimos aqueles que tem a cor de w com a cor de v.

$$C_2'(u) \triangleq \begin{cases} C_2(v) \text{ se } C_2(u) = C_2(w) \\ C_2(w) \text{ se } C_2(u) = C_2(v) \\ C_2(u) \text{ caso contrário} \end{cases}$$

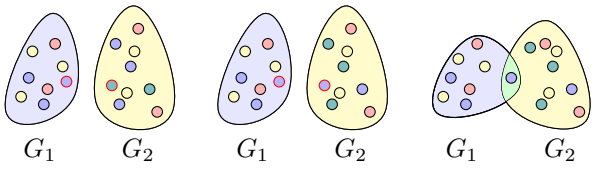


Figura 9: Identificação de vértices de cores distintas.

A coloração C é construída pelo mesmo método do caso I., mas desta vez a partir de C_1 e C_2'. Recolorir o grafo segundo este método eventualmente introduz uma nova cor ao grafo recolorido, ao custo da exclusão de uma de suas cores, fazendo com que o número de cores utilizadas seja preservado.

Portanto, este segundo caso é reduzido ao primeiro, para o qual já temos um argumento para afirmar que $\chi(\sigma_{v,w}(G_1, G_2)) = \max\{\chi(G_1), \chi(G_2)\}$.

Questão 3.:

Seja $S \subseteq \mathbb{R}^2$ um conjunto de n pontos sobre o plano. Queremos encontrar os dois pontos mais próximos entre si.

Uma abordagem simples para o problema consiste em verificar a distância entre cada um dos $\frac{n(n-1)}{2}$ pares. No entanto, uma solução em $O(n^2)$ não nos interessa, mas sim uma que seja de ordem $O(n(\log n)^2)$. Procuramos, portanto, um processo ligeiramente melhor que a solução exaustiva.

Algoritmo

Uma ideia importante é tentar evitar que a busca realize comparações entre todos os pontos do plano. É desejável comparar apenas pontos que estiverem próximos. Uma maneira natural de fazer isso é dividindo o conjunto S em duas partes iguais[2]. A princípio, como os pontos podem estar dispostos de qualquer maneira, não temos como determinar uma linha divisória do ponto de vista geométrico. Vamos, contudo, separar S de maneira que metade dos pontos estejam em S_1 e o restante em S_2.

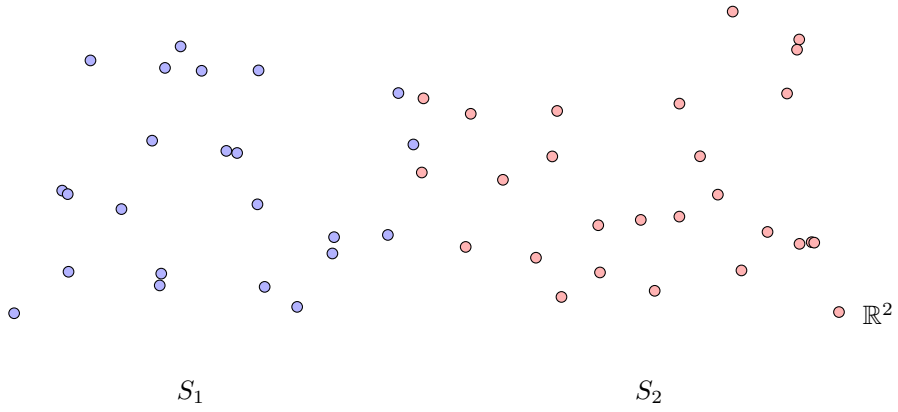


Figura 10: Dividir e conquistar.

Para tal, basta ordenar o conjunto com base na coordenada x de cada ponto. Repetiremos o processo de subdivisão em cada hemisfério e diremos que a menor distância entre dois pontos no plano é o menor dos resultados encontrados para cada subconjunto. Chamaremos $\delta(S)$ a distância entre o par de pontos mais próximos em S. $\delta(S) = \min\{\delta(S_1), \delta(S_2)\}$.

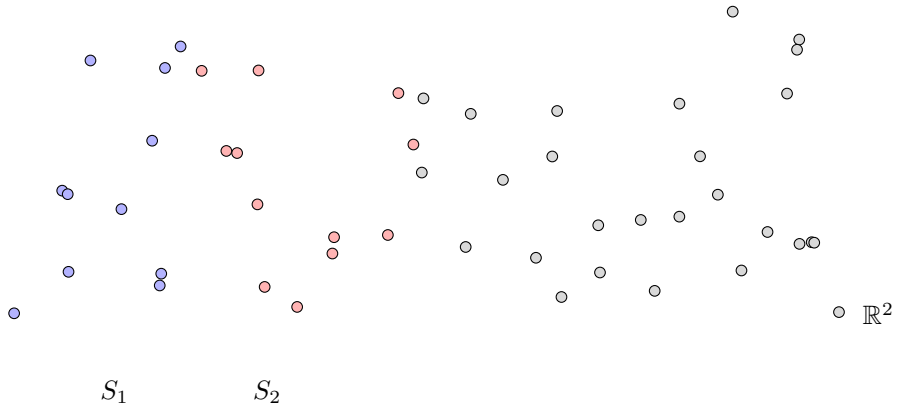


Figura 11: Dividir, dividir e conquistar.

Temos até aqui um método muito interessante e que há de se mostrar eficaz em breve. No entanto, estamos deixando escapar uma questão relevante. Na fronteira entre S_1 e S_2, é provável que exista um par de pontos p, q com $||p-q|| < \min\{\delta(S_1), \delta(S_2)\}$. Isso nos faz atentar para a necessidade de avaliar ainda um terceiro conjunto, compreendido numa faixa de largura $2\delta(S)$ ao redor da fronteira entre os conjuntos.

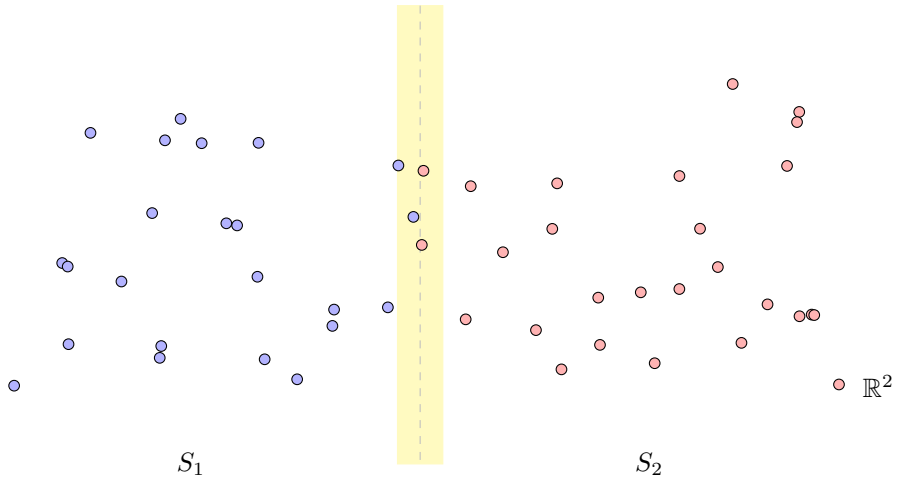


Figura 12: Dividir e conquistar com cautela.

O cálculo do par mais próximo em cada metade faz com que não seja preciso comparar pontos que estão no mesmo conjunto durante a verificação ao centro. Logo, só precisamos calcular as distâncias entre pontos de conjuntos diferentes.

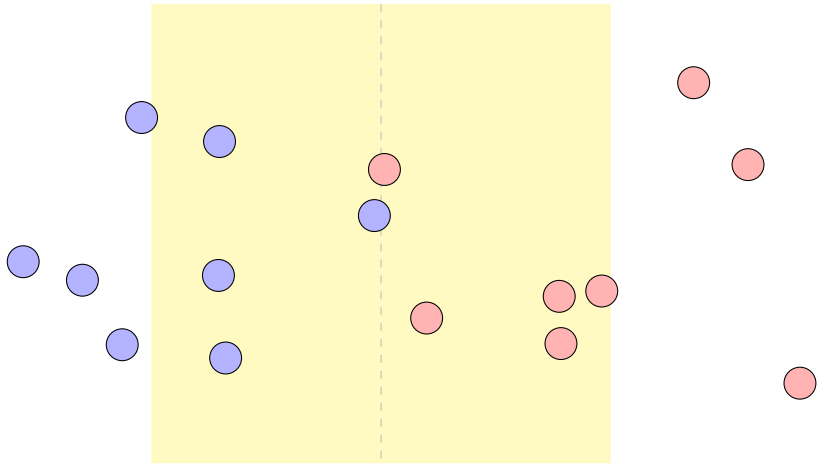


Figura 13: Faixa central.

Teorema. *Em um retângulo de lados r e $2r$ cabem, no máximo,* 8 *pontos cuja distância entre si é maior ou igual a r.*

Demonstração. Dividimos um retângulo de lados r e $2r$ em 8 quadrados cujo lado mede $\frac{r}{2}$.

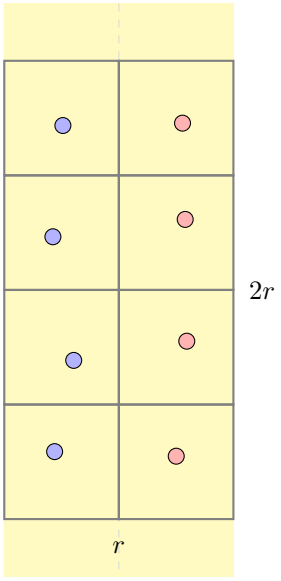


Figura 14: Faixa central.

Pelo *Princípio da Gaiola de Pombos*, se temos 8 quadrados (gaiolas) e mais do que 8 pontos (pombos), então um dos quadrados necessariamente contém mais do que um ponto. Se dois pontos ou mais se encontram em uma mesmo quadrado, então a distância entre eles deve ser menor ou igual a $\frac{r}{\sqrt{2}} < r$.

■

O resultado do teorema nos dá uma vantagem importante durante o cálculo da distância mínima na região central. Ordenando estes pontos segundo sua coordenada em y, teremos o valor desejado ao verificar somente os 8 próximos pontos na sequência, pois temos a garantia de que não encontraremos distância menor que δ se não a encontramos entre estes 8 primeiros. Esta simples observação torna o processo trivial de verificação, apresentado como $O(n^2)$ em um método linear $O(n)$, já que a verificação pode ser feita em tempo constante $O(8) = O(1)$ para cada ponto na faixa central.

Análise da Complexidade

Partindo de um conjunto S de tamanho n, precisamos ordená-lo pelas suas coordenadas x e y, obtendo dois vetores que contêm os mesmos pontos, mas em ordens provavelmente distintas. O custo de ordenação é $O(n \log n)$.

Em seguida, temos uma chamada de custo $T(n)$, que separa os dois conjuntos de vértices e realiza chamadas recursivas sobre os dois grupos de tamanho $\frac{n}{2}$, agregando $2T\left(\frac{n}{2}\right)$ ao custo de $T(n)$. Além disso, $T(n)$ ainda conta com a seleção dos pontos que pertencem a faixa central. Queremos que estes pontos estejam ordenados pela coordenada y. Como isso foi feito no início do algoritmo, podemos simplesmente filtrar os pontos p onde $|p_x - \hat{p}_x| < \delta$ vale. Isso é feito em tempo linear $O(n)$, já que visitamos cada um dos n pontos do escopo atual. O ponto $\hat{p}$ é calculado no momento da divisão dos conjuntos (ordenados em x) e é o mais próximo do $\frac{n}{2}$-ésimo elemento da lista. Por fim, verificar a proximidade na fronteira entre os conjuntos apresenta custo $O(n)$, como instrui o teorema apresentado. O processo recursivo para quando temos menos do que 3 pontos no conjunto. Com 2 pontos, digamos p e q, $\delta = ||p - q||$. Com menos do que dois pontos $\delta = \infty$. Logo, $T(2) = T(1) = 1$.

Agora, resolvemos a relação de recorrência $T(n) = 2T\left(\frac{n}{2}\right) + O(n) + O(n)$.

$$
\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + O(n) + O(n) = 2T\left(\frac{n}{2}\right) + O(n) \\
2T\left(\frac{n}{2}\right) &= 2\left(2T\left(\frac{n}{4}\right) + O\left(\frac{n}{2}\right)\right) = 4T\left(\frac{n}{4}\right) + O(n) \\
4T\left(\frac{n}{4}\right) &= 4\left(2T\left(\frac{n}{8}\right) + O\left(\frac{n}{4}\right)\right) = 8T\left(\frac{n}{8}\right) + O(n) \\
&\vdots \\
2^{k-2}T\left(\frac{n}{2^{k-2}}\right) &= 2^{k-2}\left(2T\left(\frac{n}{2^{k-1}}\right) + O\left(\frac{n}{2^{k-2}}\right)\right) = 2^{k-1}T\left(\frac{n}{2^{k-1}}\right) + O(n) \\
2^{k-1}T\left(\frac{n}{2^{k-1}}\right) &= 2^{k-1}\underbrace{T(2)}_{1} = 2^{k-1}
\end{aligned}
$$

para $k = \log n$.

Somando todas as equações, os termos para os passos intermediários de $T(n)$ se cancelam, enquanto as operações de tempo linear $O(n)$ se acumulam.

$$
\begin{aligned}
T(n) &= 2^{k-1} + O(n)(k-1) \\
&= 2^{\log n - 1} + O(n)(\log n - 1) \\
&= \frac{n}{2} + O(n)(\log n - 1) \\
&= O(n) + O(n)(\log n - 1) \\
&= O(n)(\log n)
\end{aligned}
$$

$$
\therefore T(n) = O(n \log n)
$$

Por fim, temos um custo inicial de ordenação $O(n \log n)$ seguido de um custo de busca da ordem $O(n \log n)$. Com isso, está demonstrado que o algoritmo é de tempo $O(n \log n)$ e, portanto é de tempo $O(n(\log n)^2)$.

Referências

[1] SZWARCFITER, Jayme Luiz, **Teoria Computacional de Grafos**, 1ª edição, Rio de Janeiro, 2018.

[2] WEERDT, Mathijs de, **Algoritmiek: Divide and Conquer**, TU Delft, Holanda, 2016.

Observações

- As figuras procuram mais ilustrar os conceitos e ideias propostas do que representar os objetos de maneira geometricamente fiel.
- A expressão log é utilizada para representar o logaritmo de base 2, também escrito como $\log_2$.