Teoria dos Grafos e Computabilidade
Resumos

Pedro Marçal Lima

Contents

Chapter 1

Introdução

1.1 O Que é um Grafo?

Definition 1.1: Grafo

Um grafo é um conjunto de pontos (vértices) e suas relações (arestas):

$$G = (V, E)$$

onde:

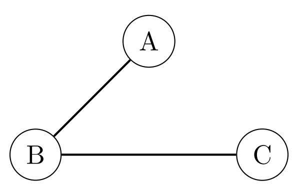
$$V = \{v_1, v_2, \ldots, v_n\}, \quad E = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots\}$$

1.2 Grafo não-direcionado

Um grafo não-direcionado, se trata de um grafo

$$G = (V, E)$$

em que suas arestas (v_i, v_j) e (v_j, v_i) são idênticas, ou seja, não importa a direção da aresta, qualquer aresta pode ser tanto de "ida" quanto de "volta".



Definição 1.2.1: Grafo Não Direcionado

$$G = (V, E)$$

onde:

$$\begin{aligned} V &= \{v_1, v_2, \ldots, v_n\} \\ E &= \{\{v_i, v_j\} \mid v_i, v_j \in V\} \\ E &\subseteq V \times V \end{aligned}$$

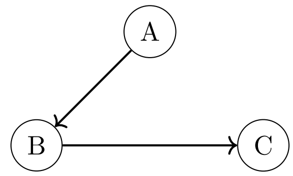
$$\{v_i, v_j\} = \{v_j, v_i\}$$

1.3 Grafo direcionado

Um grafo direcionado, se trata de um grafo

$$G = (V, E)$$

em que suas arestas (v_i, v_j) e (v_j, v_i) são diferentes, ou seja, a direção da aresta é relevante, sendo assim, considerada uma aresta de saída ou entrada em relação a um vértice.



Definição 1.3.1: Grafo Direcionado

$$G = (V, E)$$

onde:

$$V = \{v_1, v_2, \ldots, v_n\}$$
$$E = \{(v_i, v_j) \mid v_i, v_j \in V\}$$
$$E \subseteq V \times V$$

$$(v_i, v_j) \neq (v_j, v_i)$$

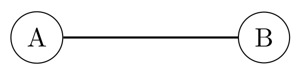
Note:-

Importante notar que, na representação de arestas, utiliza-se de {} para represenar um par não ordenado(a ordem não importa), e () para representar um par ordenado(a ordem importa)

1.4 Conceitos de vértices

1.4.1 Vértices adjacentes(vizinhos)

Vértices podem ser considerandos adjacentes ou vizinhos caso possuam uma aresta de ligação entre eles



Definition 1.2: Vértices adjacentes

Dado dois vértices:

$$v_i \text{ e } v_j$$

onde:

$$v_i \land v_j \in V$$

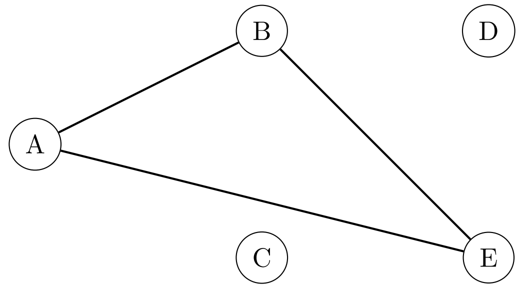
$$(v_i, v_j) \in E$$

Entende-se que:

$$v_i \leftrightarrow v_j$$

1.4.2 Vértice isolado

É um vértice v que possui $d(v) = 0$, ou seu conjunto de arestas vazio(como D e C no grafo abaixo).



Definition 1.3: Vértice Isolado

Dado um grafo

$$G = (V, E)$$

onde:

$$v \in V$$

temos que:

$$\begin{cases} \forall u \in V, \quad \{v, u\} \notin E & \text{(se } G \text{ é não direcionado)} \\ \forall u \in V, \quad (u, v) \notin E \land (v, u) \notin E & \text{(se } G \text{ é direcionado)} \end{cases}$$

ou:

$$\begin{cases} d(v) = 0 & \text{(se } G \text{ é não direcionado)} \\ d^+(v) + d^-(v) = 0 & \text{(se } G \text{ é direcionado)} \end{cases}$$

Portanto, podemos concluir que v é um vértice isolado, pois:

$$N(v) = \emptyset$$

Note:-

A expressão $N(v)$ se refere ao conjunto vizinhança de um vértice (neighborhood), ou seja, os vértices que ele possui ligação por uma aresta

1.5 Aresta incidente

Caso um vértice v, seja o vértice final de uma aresta $e = u, v$, e é incidente em v a partir de u

Definition 1.4: Aresta incidente

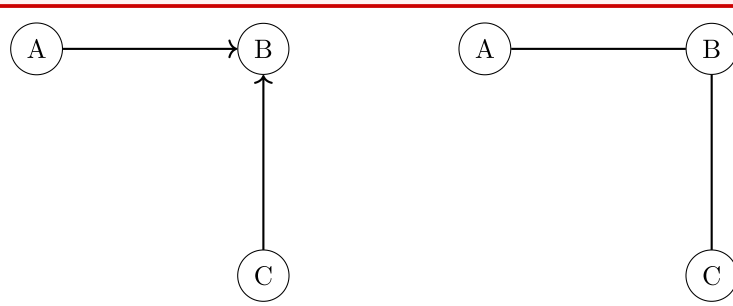
Em um grafo G = (V, E), uma aresta incidente e em v, é uma aresta $e = (u, v)$

1.6 Arestas adjacentes(vizinhas)

Arestas não paralelas podem ser consideradas adjacentes ou vizinhas caso sejam incidentes em um vértice comum

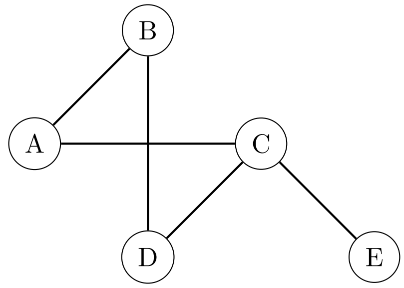
Definition 1.5: Arestas adjacentes

Em um grafo $G = (V, E)$ é um par de arestas $e_1 = (u, v)$ $e_2 = (x, v) \mid e_1 \land e_2 \in E$



1.7 Grafo Simples

Um grafo simples é um grafo que não possui nem loops nem arestas parelelas



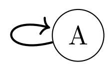
1.7.1 Loop

Um loop se trata de uma aresta associada a um par de vértices (v_i, v_i)

Definition 1.6: Loop

Considerando um vértice v_i, um loop será definido por:

$$E = \{(v_i, v_j) \mid v_i, v_j \in V \mid v_i = v_j\}$$



1.7.2 Arestas Parelelas

Arestas Paralelas ocorrem quando existe mais de uma aresta associada ao mesmo par de vértices

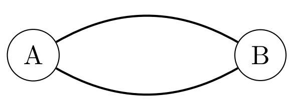
Definição 1.7.1: Arestas Paralelas

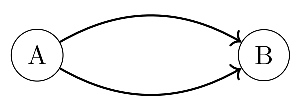
Para um Vértice V_i

$$E = \{(v_i, v_j), (v_i, v_j), \ldots\}$$

Onde E é o conjunto de arestas tal que:

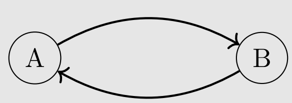
$$E \subseteq V \times V$$





Note:-

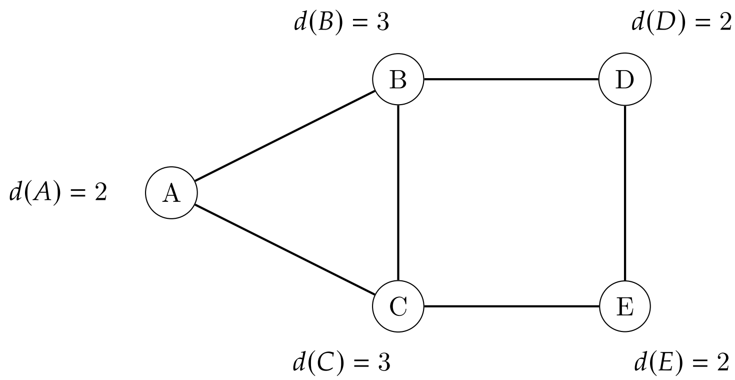
Importante ressaltar que as arestas para serem consideradas paralelas em um grafo direcionado, devem possuir o mesmo sentido/direção
Exemplo de grafo sem arestas parelelas:



1.8 Graus

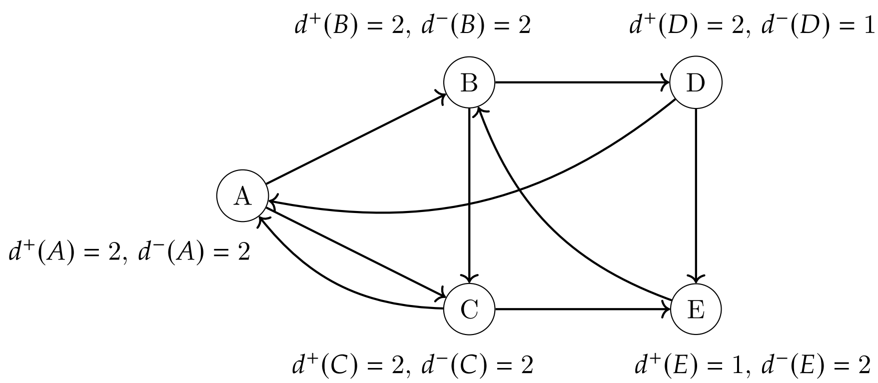
1.8.1 Grafo não direcionado

Em um grafo não direcionado G =(V, E) , a quantidade de arestas em um vértice determina o seu grau $d(v)$



1.8.2 Grafo direcionado

Em um grafo direcionado, existem dois tipos de graus, os de saída, que se referem as arestas que saem do determinado vértice $d^{+}(v)$, e de entrada, que se referem as arestas que chegam ao vértice $d^{-}(v)$



1.9 Fecho Transitivo Direto

Para se falar de fecho transitivo direto, é a mesma coisa que falar da atingibilidade de um vértice, ou seja, quem é alcançavél a partir de um vértice v

Definition 1.7: Fecho Transitivo Direto

Dado um grafo $G = (V, E)$
o seu fecho = v — vi, vj C E

1.10 Caminho

Caminho é uma sequência de vértices e arestas que começa e termina em um determinado vértice, que casso possua determinada características, possuirá um nome diferente, que são:

1.10.1 Walk (Caminho)

Um caminho é considerado "Walk", para qualquer caminho dentro de um grafo, sem nenhuma restrição além da definição básica de caminho.

Definition 1.8: Walk

Dado um caminho

$$P = v_1, e_1, v_2, e_2, \ldots, v_k$$

onde:

$$v_i \in V, \quad e_i \in E,$$

conclui-se que:

$$P = W$$

1.10.2 Trail (Trilha)

Um caminho é considerado "Trail", caso não possua repetição de arestas

Definition 1.9: Trail

Dado um caminho:

$$P = v_1, e_1, v_2, e_2, \ldots, v_k,$$

onde:

$$v_i \in V, \quad e_i \in E,$$

se:

$$e_i \neq e_j, \forall i \neq j,$$

então:

$$P = T.$$

1.10.3 Path (Caminho simples)

Um caminho é considerado Path, caso não possua repetição nem de vértices nem de arestas

Definition 1.10: Path

Dado um caminho:

$$P = v_1, e_1, v_2, e_2, \ldots, v_k,$$

onde:

$$v_i \in V, \quad e_i \in E,$$

se:

$$e_i \neq e_j, \forall i \neq j,$$

e:

$$v_i \neq v_j, \forall i \neq j,$$

então:

$$P = P_{simples}.$$

1.11 Isomorfismo

Dois grafos G e H são considerados isomorfos caso seja possível fazer uma relação de vértice para vértice e suas arestas entre todos os vértice de V, com as incidências também se mantendo

Definition 1.11: Isomorfismo

Dado dois grafos

$$G = (V, E) \text{ e } H = (V, E)$$

que possuem:

$$\text{Seq}_G = (d(u_1), d(u_2), \ldots, d(u_n))$$
$$\text{Seq}_H = (d(u_1), d(u_2), \ldots, d(u_n))$$

com as seguintes propriedades:

$$|V_G| = |V_H|$$
$$|E_G| = |E_H|$$
$$\text{Seq}_G = \text{Seq}_H$$
$$f : V_G \to V_H$$

tal que, para quaisquer vértices $u, v \in V_G$, temos:

$$\{u, v\} \in E_G \iff \{f(u), f(v)\} \in E_H.$$

apesar de não ser o suficiente para garantir, conclui-se que existe a possibilidade de:

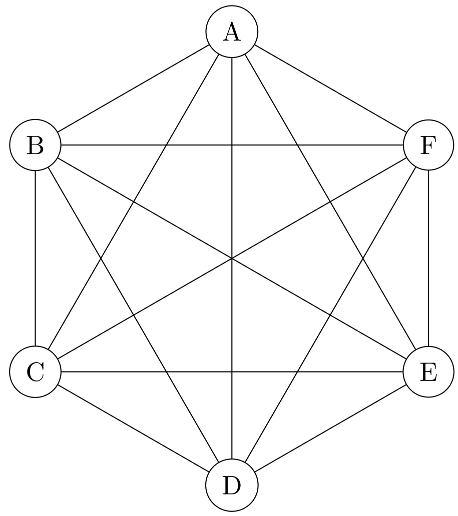
$$G \cong H$$

Note:-

Essa definição apresenta propriedades necessárias, mas não suficientes, para garantir o isomorfismo. Mesmo que a correspondência dos vértices, arestas e graus seja necessária, não é suficiente para concluir o isomorfismo sem a condição de preservação de adjacência(relação entre arestas).

1.12 Grafo Completo

Um grafo completo é um grafo que possui em seu conjunto de arestas E todas arestas possíveis do grafo.



Definition 1.12: Grafo Completo

Dado um grafo

$$G = (V, E)$$

onde:

$$|\, E \,| = \frac{n \cdot (n-1)}{2}$$

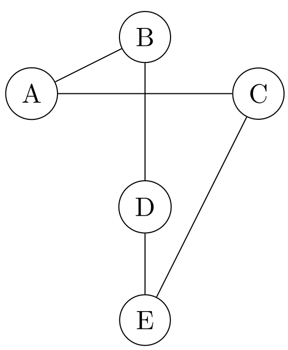
$$\forall v \in V, \quad d(v) = n - 1$$

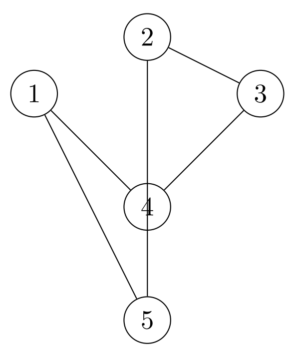
Conclui-se que:

$$G = K_n$$

1.13 Grafo Complementar

Um grafo complementar de um Grafo $G = (V, E)$ é um grafo $G' = (V, E')$ em que suas arestas correpondem a todas arestas faltantes em G para se ter um grafo completo.





Definition 1.13: Grafo Complementar

Dado dois grafos

$$G = (V, E) \text{ e } G' = (V, E')$$

onde:

$$E \cap E' = \emptyset$$

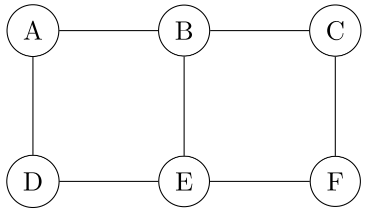
$$|E| + |E'| = \frac{n \cdot (n-1)}{2}$$

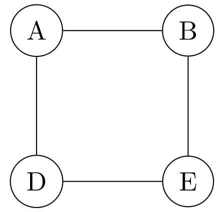
conclui-se que:

$$E' = \overline{E}$$

1.14 Subgrafo

Dizemos que um Grafo é subgrafo de outro grafo, caso todos elementos(vértices e arestas) do subgrafo façam parte do outro Grafo





Definition 1.14: Subgrafo

Dado dois grafos

$$G = (V, E) \text{ e } H = (V, E)$$

onde:

$$V_H \subseteq V_G$$

$$E_H \subseteq E_G$$

ou:

$$\forall e \in E_H,\, e \in E_G$$

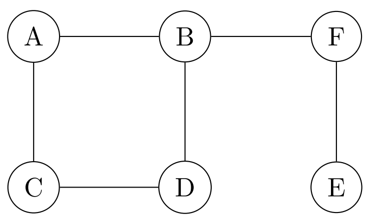
$$\forall v \in V_H,\, v \in V_G$$

Conclui-se que:

$$H \subseteq G$$

1.15 Ciclos

Um grafo possui ciclos caso caso possua um caminho que possui o mesmo vértice de início e fim Por exemplo o caminho: $P = A, AC, C, CD, D, DB, B, BA, A$



Definition 1.15: Ciclo

Dado um grafo

$$G = (V, E)$$

onde:

$$P = v_1, e_1, v_2, e_2, v_3, e_3, \ldots, v_k, e_k, v_1$$

$$V' = \{v_2, v_3, \ldots, v_k\} \text{ (todos vértices menos o } v_1\text{)}$$

$$v_i \neq v_j,\ \forall v_i, v_j \in V',\ i \neq j$$

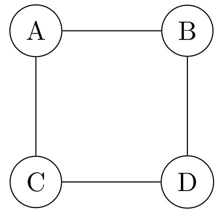
$$e_i \neq e_j,\ \forall e_i, e_j \in E,\ i \neq j$$

então:

$$P = C$$

1.16 Grafo Ciclo

Um grafo Ciclo é um grafo que possui um caminho ciclo com todos seus vértices, ou seja, ele inteiro é um ciclo.



Definition 1.16: Grafo Ciclo

Dado um grafo

$$G = (V, E)$$

onde:

$$\forall v \in V,$$

$$\mathcal{P}_{\max}(v) = \underset{\mathcal{P} \in \mathcal{P}(v)}{\arg\max} |E(\mathcal{P})| \ \land \ \mathcal{P}_{\max}(v) = C$$

então:

$$\mathcal{P}_{\max}(v) = C_n$$

1.17 Termos

Chapter 2

Algortimos

2.1 Busca em profundidade(DFS)

A busca em profundidade

Chapter 3

3.1 Random Examples

Definição 3.1.1: Limit of Sequence in $\mathbb{R}$

Let $\{s_n\}$ be a sequence in $\mathbb{R}$. We say

$$\lim_{n\to\infty} s_n = s$$

where $s \in \mathbb{R}$ if $\forall$ real numbers $\epsilon > 0$ $\exists$ natural number N such that for $n > N$

$$s - \epsilon < s_n < s + \epsilon \text{ i.e. } |s - s_n| < \epsilon$$

Questão 1

Is the set $x-$axis$\setminus\{$Origin$\}$ a closed set

Solução: We have to take its complement and check whether that set is a open set i.e. if it is a union of open balls

Note:-

We will do topology in Normed Linear Space (Mainly $\mathbb{R}^n$ and occasionally $\mathbb{C}^n$)using the language of Metric Space

Claim 3.1.1 Topology

Topology is cool

Example 3.1.1 (Open Set and Close Set)

Open Set:
- ϕ
- $\bigcup\limits_{x\in X} B_r(x)$ (Any $r > 0$ will do)
- $B_r(x)$ is open

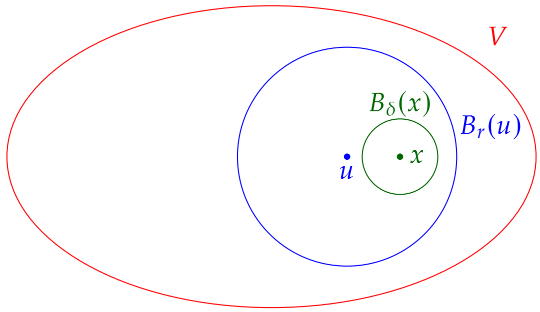
Closed Set:
- $X,\ \phi$
- $\overline{B_r(x)}$

$x-$axis $\cup$ $y-$axis

Theorem 3.1.1

If $x \in$ open set V then $\exists\ \delta > 0$ such that $B_\delta(x) \subset V$

Proof: By openness of V, $x \in B_r(u) \subset V$



Given $x \in B_r(u) \subset V$, we want $\delta > 0$ such that $x \in B_\delta(x) \subset B_r(u) \subset V$. Let $d = d(u, x)$. Choose δ such that $d + \delta < r$ (e.g. $\delta < \frac{r-d}{2}$)

If $y \in B_\delta(x)$ we will be done by showing that $d(u, y) < r$ but

$$d(u, y) \leqslant d(u, x) + d(x, y) < d + \delta < r$$

😄

Corollary 3.1.1
By the result of the proof, we can then show...

Lenma 3.1.1
Suppose $\vec{v}_1, \ldots, \vec{v}_n \in \mathbb{R}^n$ is subspace of $\mathbb{R}^n$.

Proposition 3.1.1
$1 + 1 = 2$.

3.2 Random

Definição 3.2.1: Normed Linear Space and Norm $\|\cdot\|$

Let V be a vector space over $\mathbb{R}$ (or $\mathbb{C}$). A norm on V is function $\|\cdot\|$ $V \longrightarrow \mathbb{R}_{\geqslant 0}$ satisfying

1. $\|x\| = 0 \iff x = 0 \; \forall \; x \in V$
2. $\|\lambda x\| = |\lambda| \|x\| \; \forall \; \lambda \in \mathbb{R}(\text{or } \mathbb{C}),\; x \in V$
3. $\|x + y\| \leqslant \|x\| + \|y\| \; \forall \; x, y \in V$ (Triangle Inequality/Subadditivity)

And V is called a normed linear space.
• Same definition works with V a vector space over $\mathbb{C}$ (again $\|\cdot\| \longrightarrow \mathbb{R}_{\geqslant 0}$) where 2 becomes $\|\lambda x\| = |\lambda| \|x\|$ $\forall \; \lambda \in \mathbb{C}$, $x \in V$, where for $\lambda = a + ib$, $|\lambda| = \sqrt{a^2 + b^2}$

Example 3.2.1 ($p-$Norm)
$V = \mathbb{R}^m$, $p \in \mathbb{R}_{\geqslant 0}$. Define for $x = (x_1, x_2, \cdots, x_m) \in \mathbb{R}^m$

$$\|x\|_p = \left(|x_1|^p + |x_2|^p + \cdots + |x_m|^p\right)^{\frac{1}{p}}$$

(In school $p = 2$)

Special Case $p = 1$: $\|x\|_1 = |x_1| + |x_2| + \cdots + |x_m|$ is clearly a norm by usual triangle inequality.

Special Case $p \to \infty$ ($\mathbb{R}^m$ with $\|\cdot\|_\infty$): $\|x\|_\infty = \max\{|x_1|, |x_2|, \cdots, |x_m|\}$
For $m = 1$ these $p-$norms are nothing but $|x|$. Now exercise

Questão 2

Prove that triangle inequality is true if $p \geqslant 1$ for $p-$norms. (What goes wrong for $p < 1$?)

Solução: **For Property ③ for norm-2**

When field is $\mathbb{R}$:

We have to show

$$\sum_i (x_i + y_i)^2 \leqslant \left(\sqrt{\sum_i x_i^2} + \sqrt{\sum_i y_i^2}\right)^2$$

$$\implies \sum_i (x_i^2 + 2x_iy_i + y_i^2) \leqslant \sum_i x_i^2 + 2\sqrt{\left[\sum_i x_i^2\right]\left[\sum_i y_i^2\right]} + \sum_i y_i^2$$

$$\implies \left[\sum_i x_iy_i\right]^2 \leqslant \left[\sum_i x_i^2\right]\left[\sum_i y_i^2\right]$$

So in other words prove $\langle x, y\rangle^2 \leqslant \langle x, x\rangle\langle y, y\rangle$ where

$$\langle x, y\rangle = \sum_i x_iy_i$$

Note:-

- $\|x\|^2 = \langle x, x\rangle$
- $\langle x, y\rangle = \langle y, x\rangle$
- $\langle\cdot,\cdot\rangle$ is $\mathbb{R}-$linear in each slot i.e.

$$\langle rx + x', y\rangle = r\langle x, y\rangle + \langle x', y\rangle \text{ and similarly for second slot}$$

  Here in $\langle x, y\rangle$ x is in first slot and y is in second slot.

Now the statement is just the Cauchy-Schwartz Inequality. For proof

$$\langle x, y\rangle^2 \leqslant \langle x, x\rangle\langle y, y\rangle$$

expand everything of $\langle x - \lambda y, x - \lambda y\rangle$ which is going to give a quadratic equation in variable λ

$$\begin{aligned}\langle x - \lambda y, x - \lambda y\rangle &= \langle x, x - \lambda y\rangle - \lambda\langle y, x - \lambda y\rangle \\ &= \langle x, x\rangle - \lambda\langle x, y\rangle - \lambda\langle y, x\rangle + \lambda^2\langle y, y\rangle \\ &= \langle x, x\rangle - 2\lambda\langle x, y\rangle + \lambda^2\langle y, y\rangle\end{aligned}$$

Now unless $x = \lambda y$ we have $\langle x - \lambda y, x - \lambda y\rangle > 0$ Hence the quadratic equation has no root therefore the discriminant is greater than zero.

When field is $\mathbb{C}$:

Modify the definition by

$$\langle x, y\rangle = \sum_i \overline{x_i}y_i$$

Then we still have $\langle x, x\rangle \geqslant 0$

3.3 Algorithms

Algorithm 1: what

```
Input: This is some input
Output: This is some output
/* This is a comment */

some code here;
x ← 0;
y ← 0;
if x > 5 then
    x is greater than 5 ;                         // This is also a comment
else
    x is less than or equal to 5;
end
foreach y in 0..5 do
    y ← y + 1;
end
for y in 0..5 do
    y ← y − 1;
end
while x > 5 do
    x ← x − 1;
end
return Return something here;
```