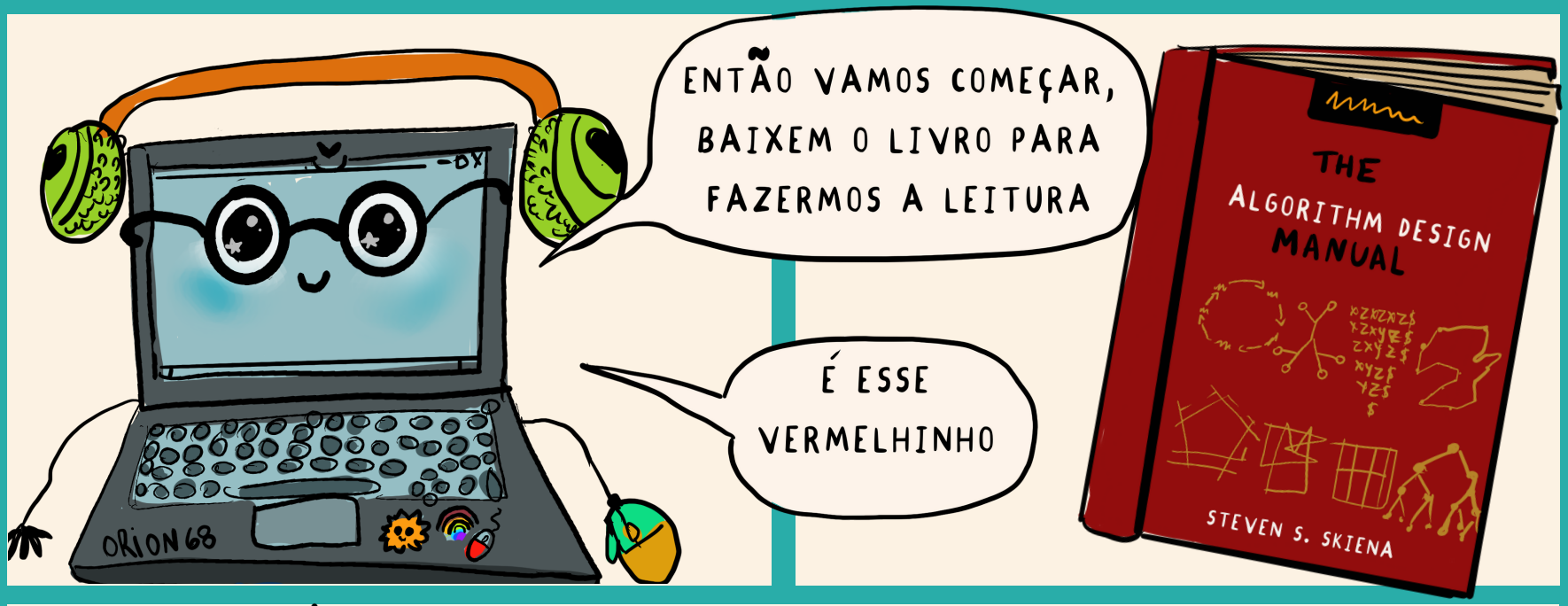
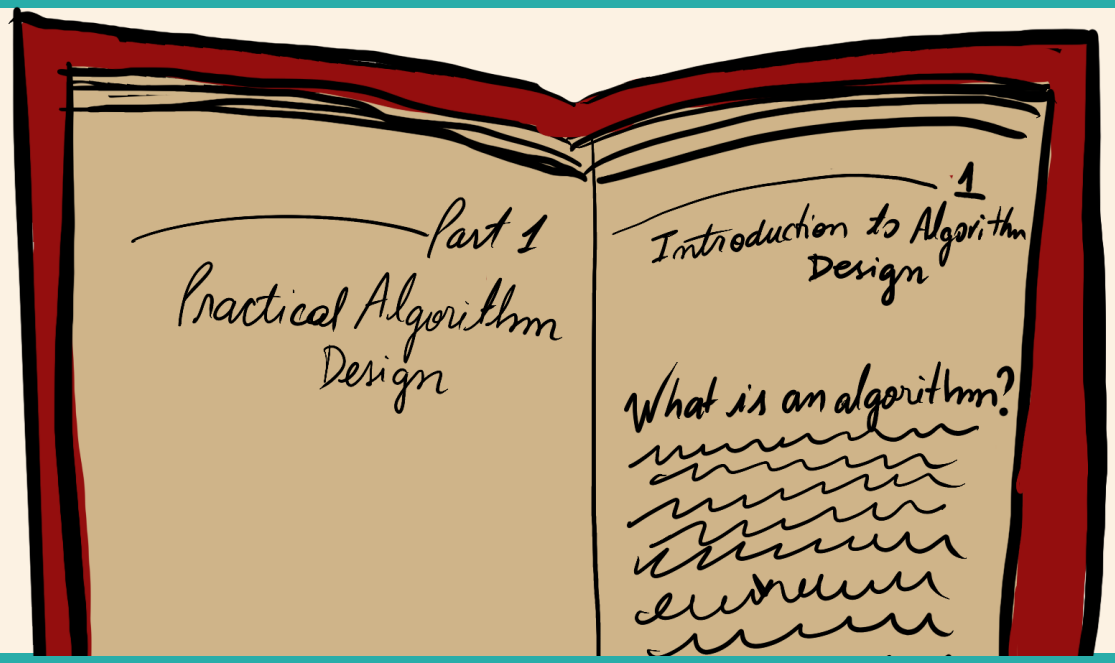
Otimização e Complexidade de Algoritmos

Estudos do Livro Vermelho

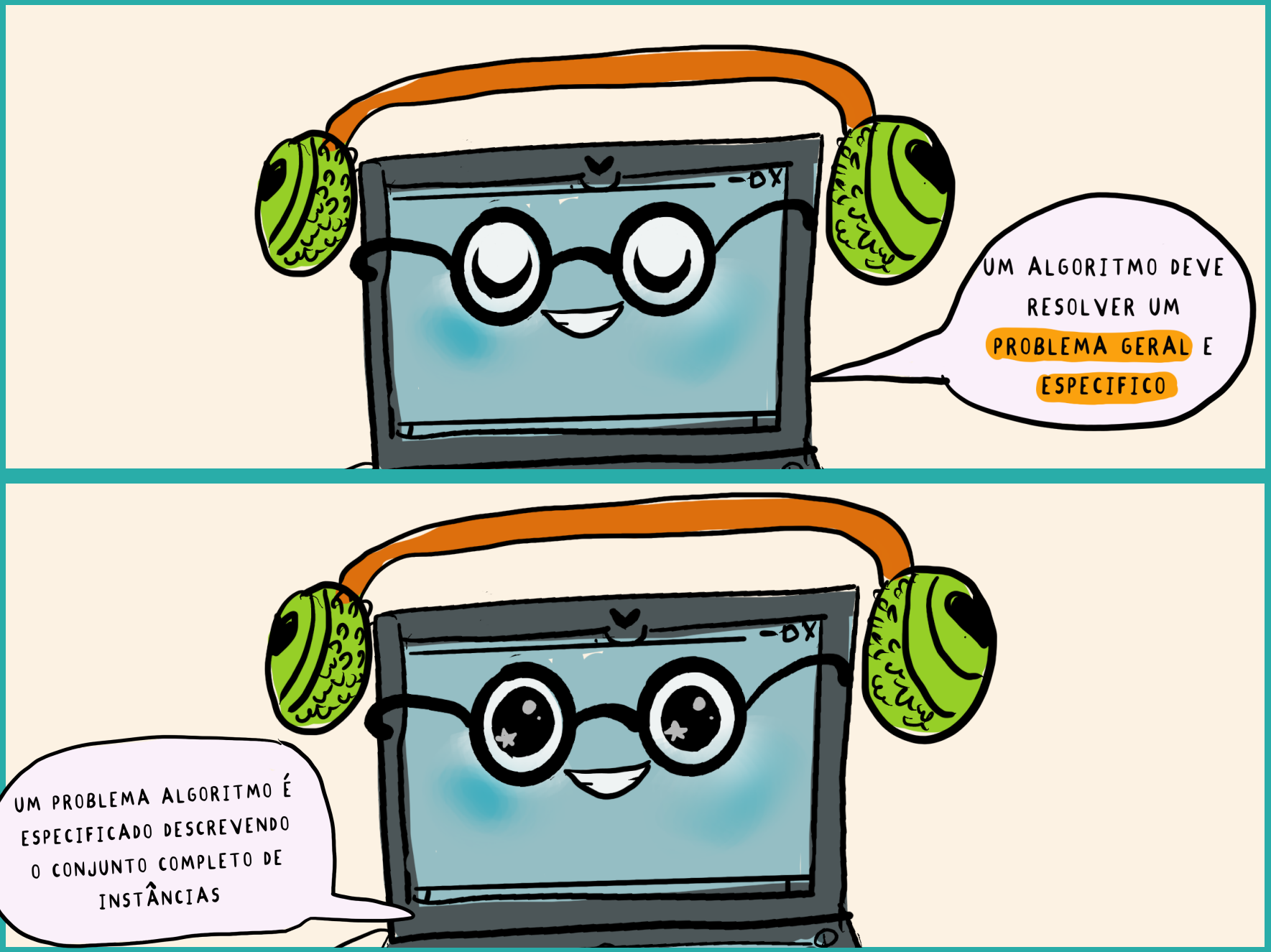
AMANDA GOBUS
DIEGO COMIS
RODRIGO CARGNELUTTI

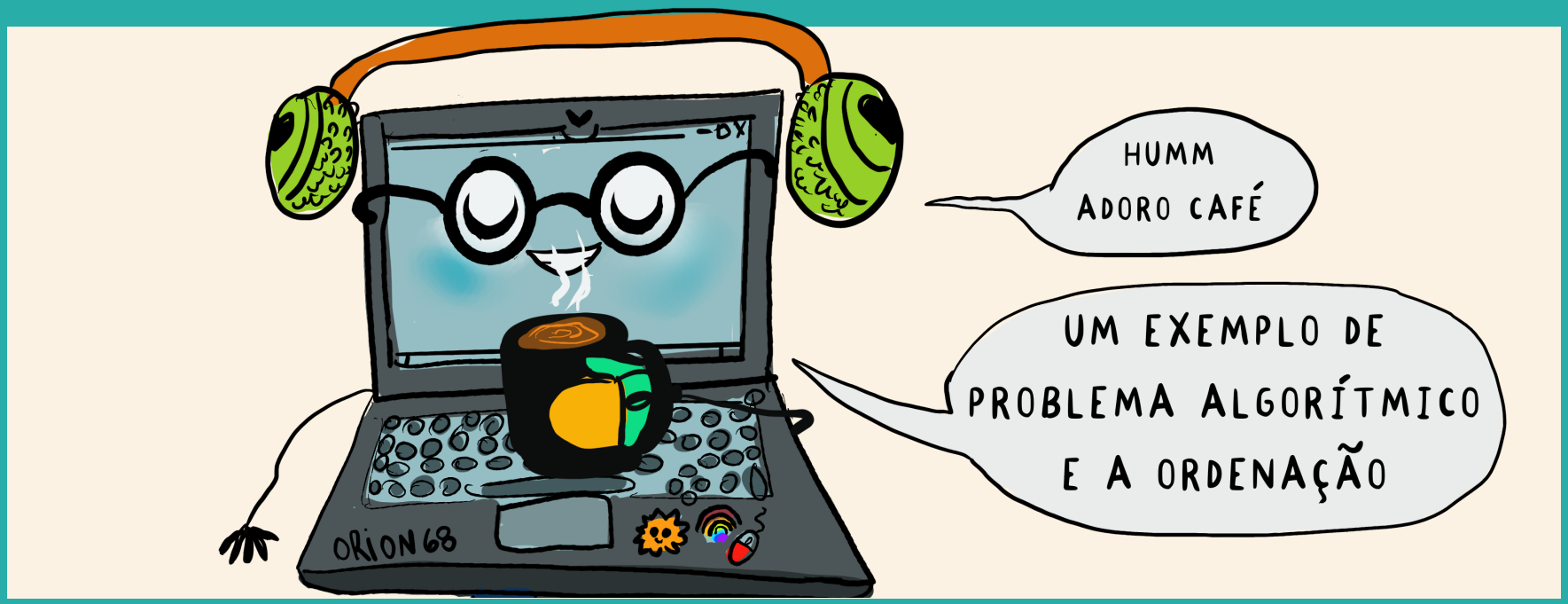
OTIMIZAÇÃO E COMPLEXIDADE DE ALGORITMOS


ENTÃO VAMOS COMEÇAR, BAIXEM O LIVRO PARA FAZERMOS A LEITURA
É ESSE VERMELHINHO
ORION 68
THE ALGORITHM DESIGN MANUAL
STEVEN S. SKIENA

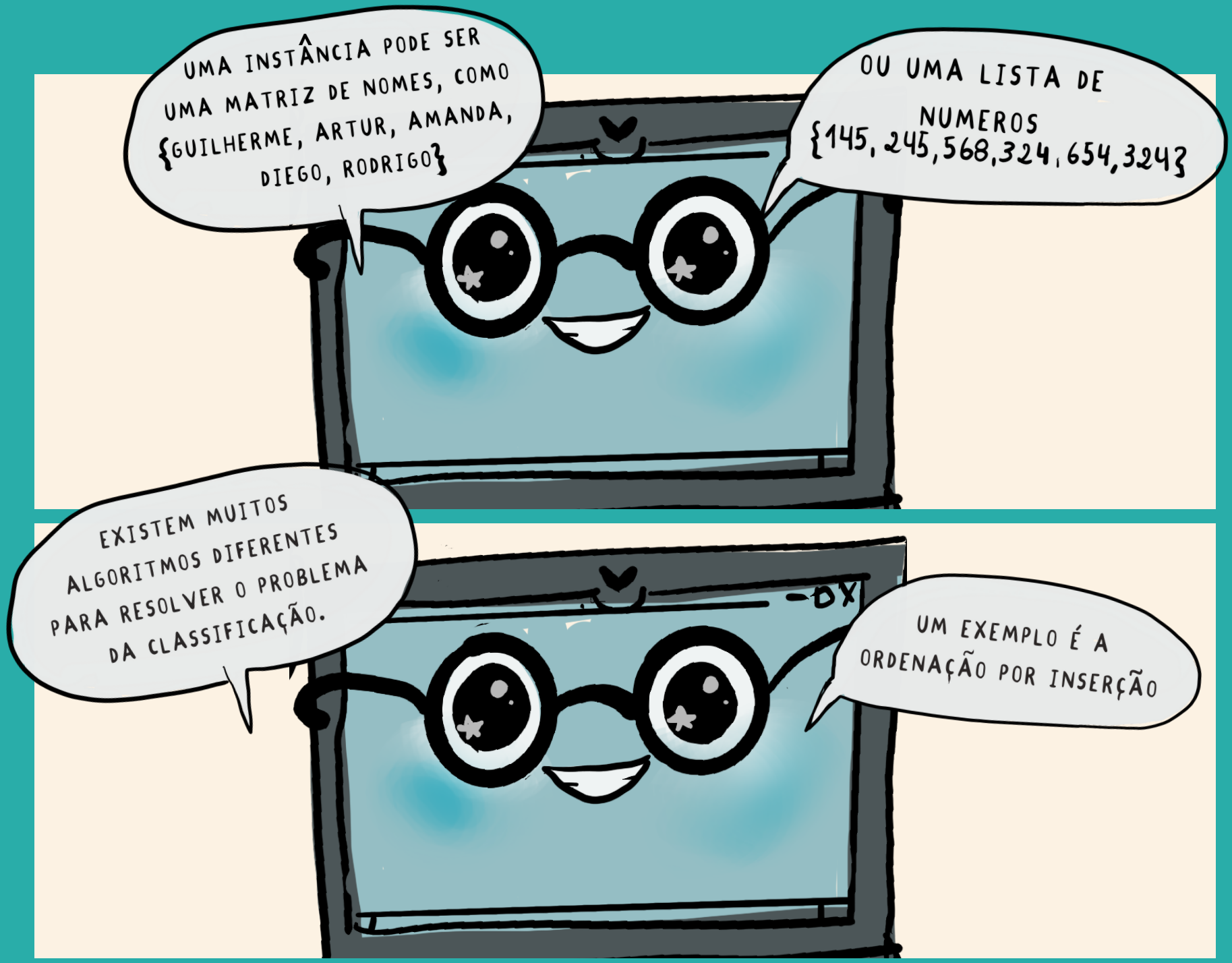

Part 1
Practical Algorithm Design
1
Introduction to Algorithm Design
What is an algorithm?

O QUE É UM ALGORITMO?
O ALGORITMO É UM PROCEDIMENTO PARA REALIZAR UMA TAREFA ESPECIFICA.
ORION 68

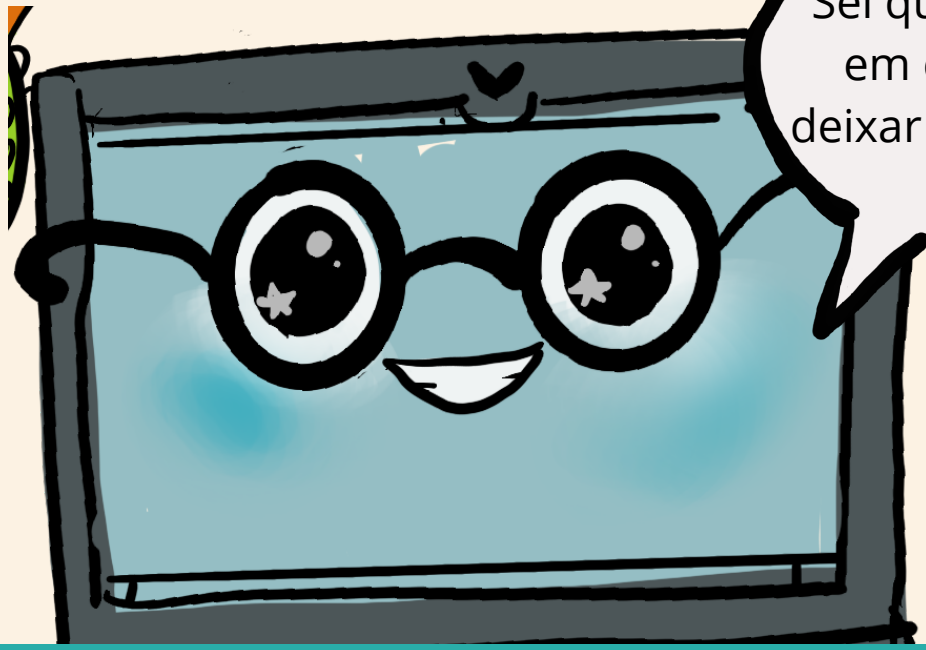

UM ALGORITMO DEVE RESOLVER UM PROBLEMA GERAL E ESPECIFICO
UM PROBLEMA ALGORITMO É ESPECIFICADO DESCREVENDO O CONJUNTO COMPLETO DE INSTÂNCIAS



Problema: Classificação
Entrada: Uma sequência de n teclas a1, ..., an.
Saída: A permutação (reordenação) da sequência de entrada tal que a1´ ≤ a2´ ≤ ... ≤ an-1´ ≤ an´

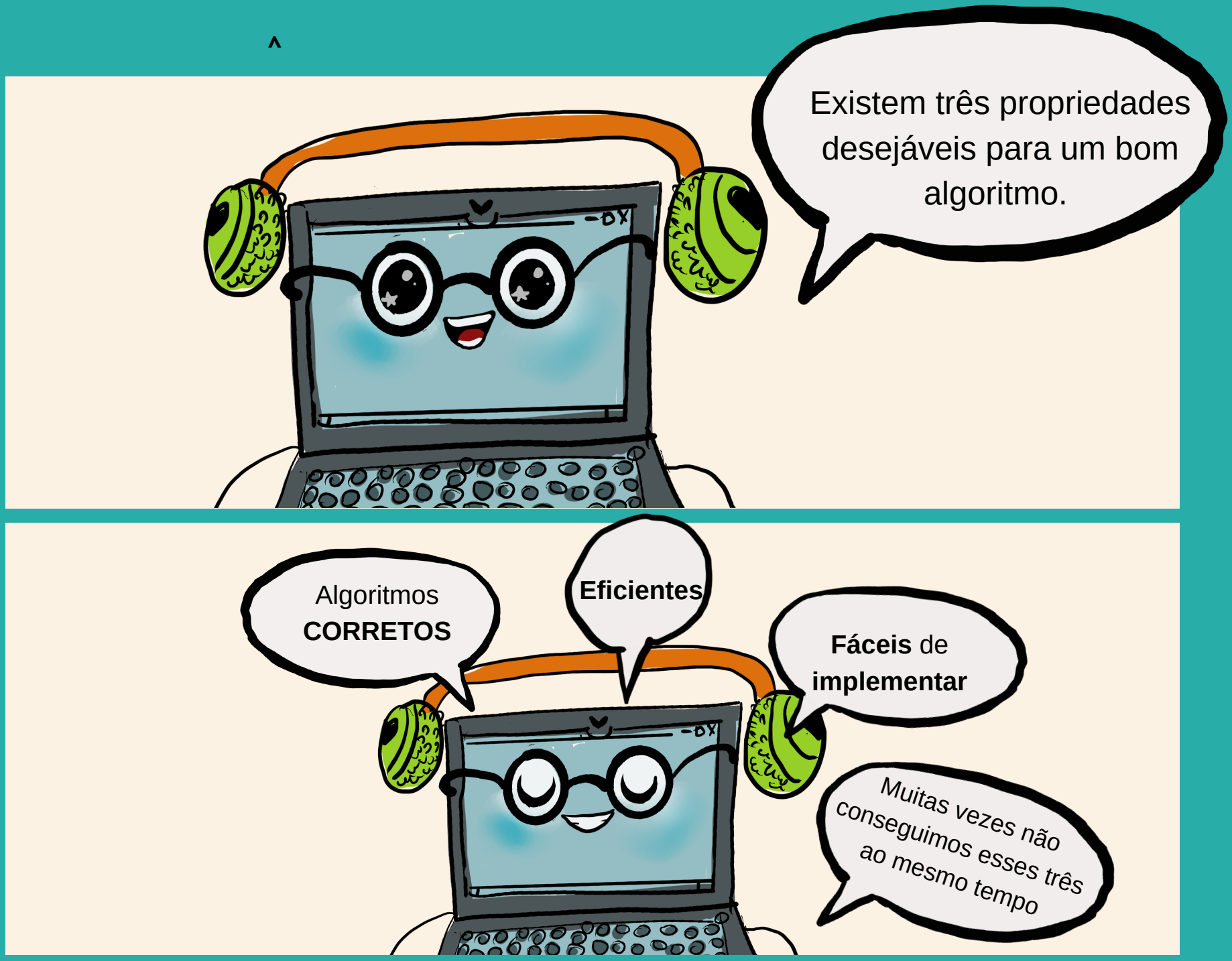

UMA INSTÂNCIA PODE SER UMA MATRIZ DE NOMES, COMO {GUILHERME, ARTUR, AMANDA, DIEGO, RODRIGO}
OU UMA LISTA DE NUMEROS {145, 245, 568, 324, 654, 324}
EXISTEM MUITOS ALGORITMOS DIFERENTES PARA RESOLVER O PROBLEMA DA CLASSIFICAÇÃO.
UM EXEMPLO É A ORDENAÇÃO POR INSERÇÃO

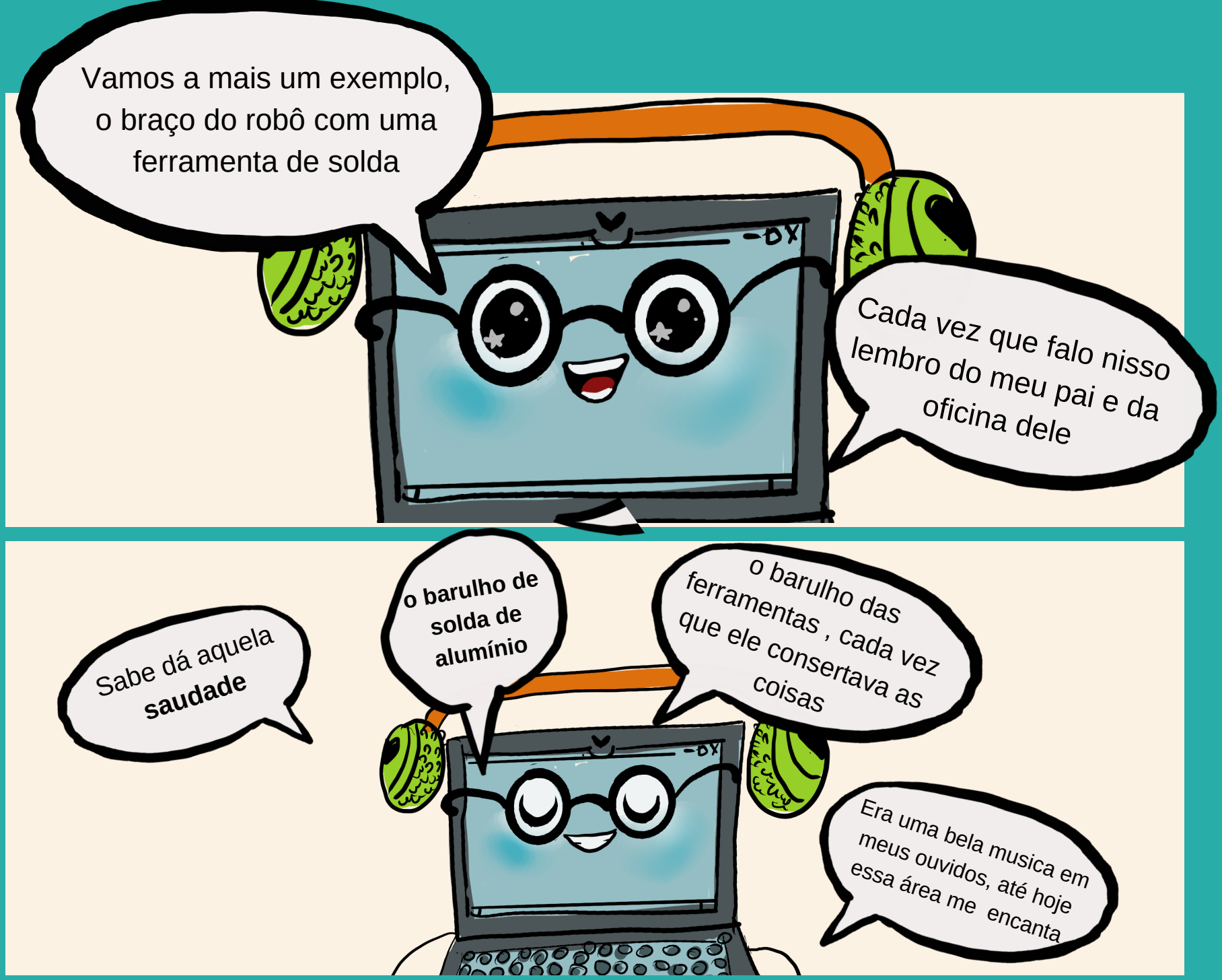
^

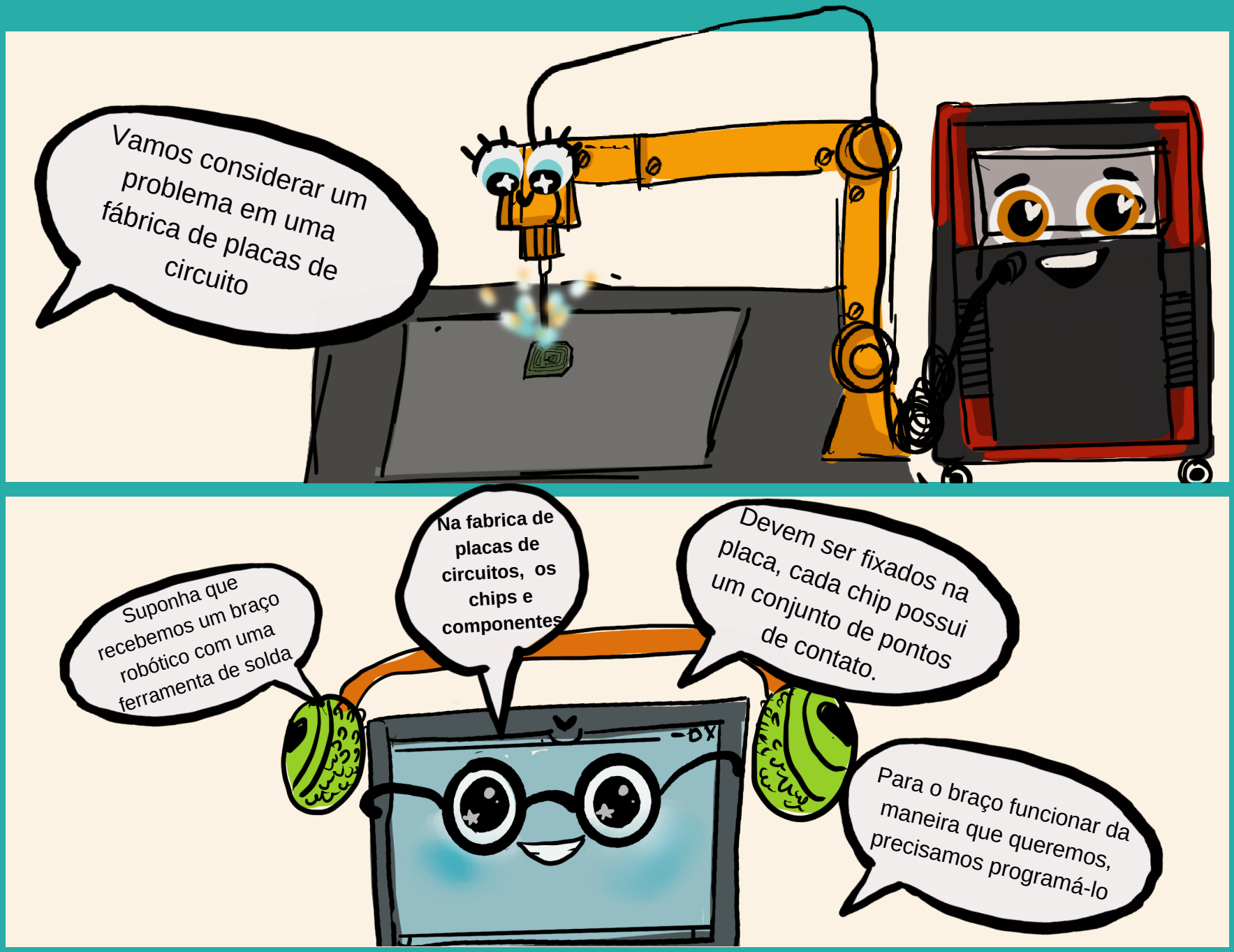


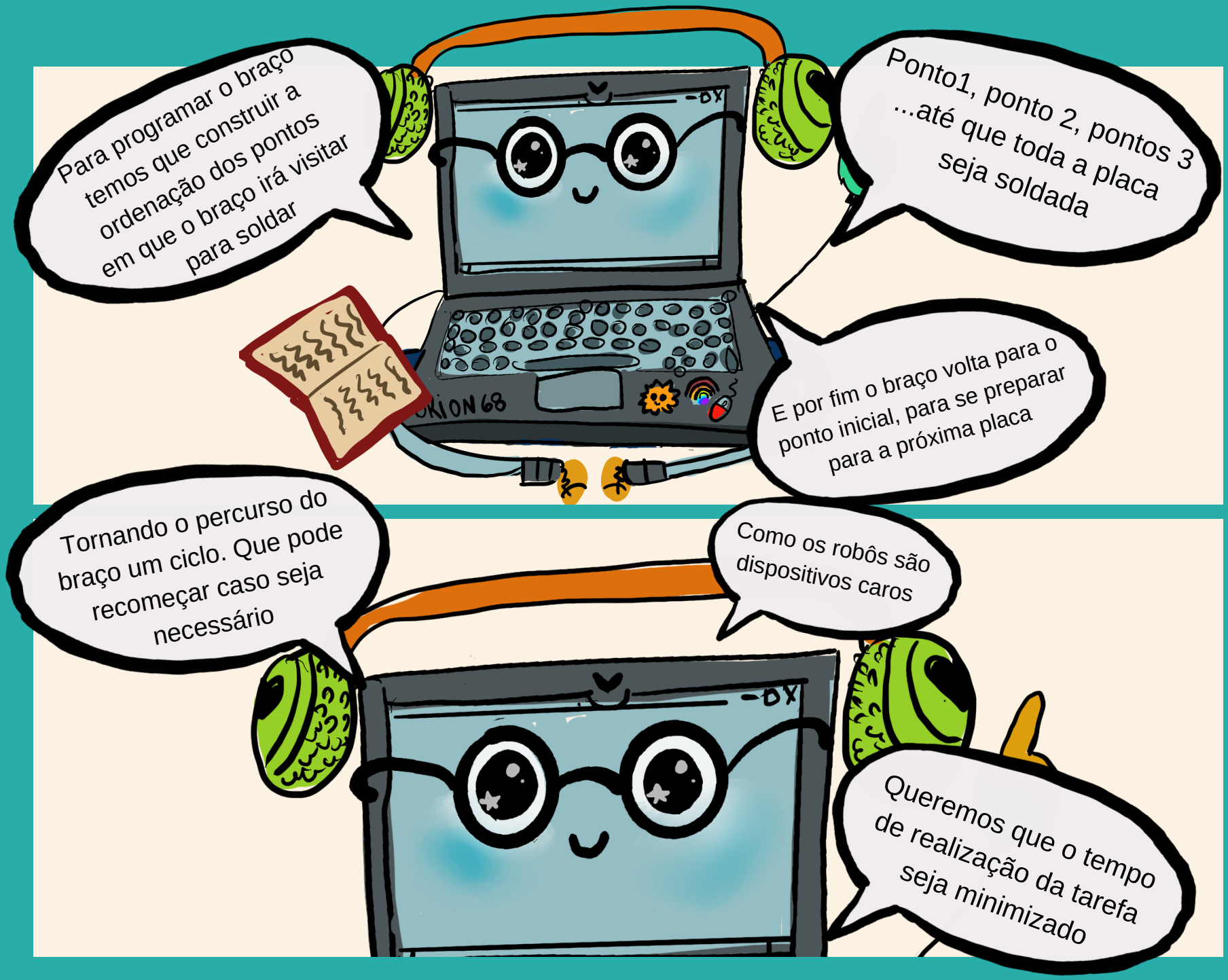

```
insertion_sort(item s[], int n)
        {
            int i,j;  /* counters */
            for (i=1; i<n; i++) {
                j=i;
                while ((j>0) && (s[j] < s[j-1])) {
                    swap(&s[j],&s[j-1]);
                    j = j-1;
            }
        }
}
```

^


Existem três propriedades desejáveis para um bom algoritmo.
Algoritmos **CORRETOS**
Eficientes
Fáceis de **implementar**
Muitas vezes não conseguimos esses três ao mesmo tempo

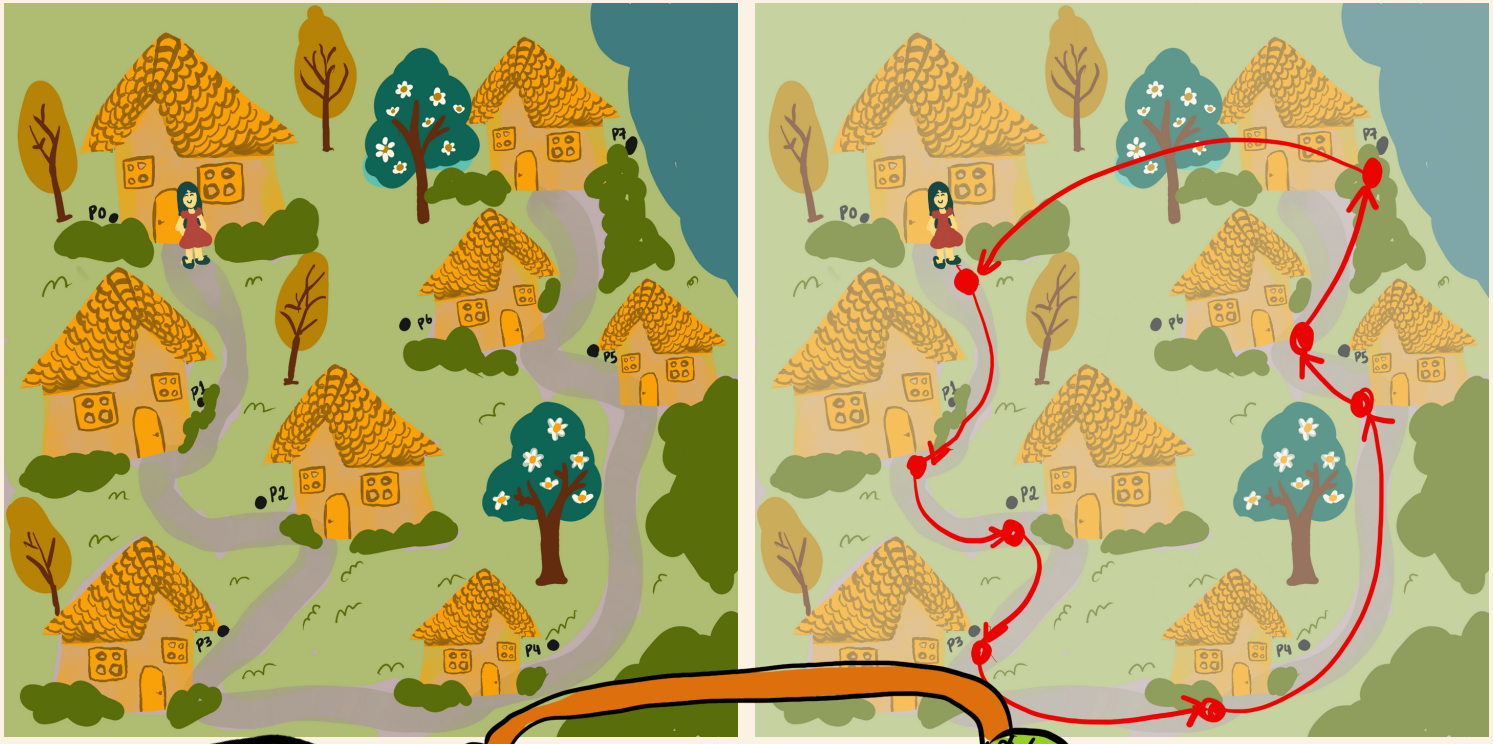

Vamos a mais um exemplo, o braço do robô com uma ferramenta de solda
Cada vez que falo nisso lembro do meu pai e da oficina dele
Sabe dá aquela **saudade**
o barulho de solda de alumínio
o barulho das ferramentas , cada vez que ele consertava as coisas
Era uma bela musica em meus ouvidos, até hoje essa área me encanta


Vamos considerar um problema em uma fábrica de placas de circuito
Suponha que recebemos um braço robótico com uma ferramenta de solda
Na fabrica de placas de circuitos, os chips e componentes
Devem ser fixados na placa, cada chip possui um conjunto de pontos de contato.
Para o braço funcionar da maneira que queremos, precisamos programá-lo

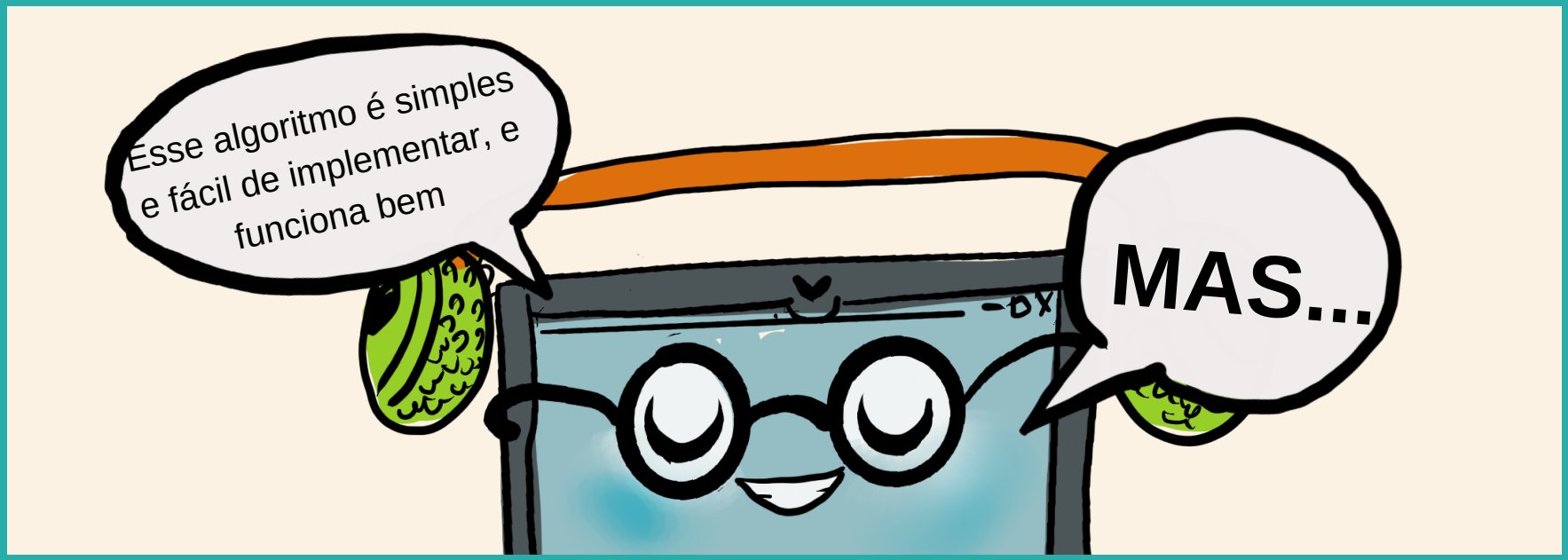

Para programar o braço temos que construir a ordenação dos pontos em que o braço irá visitar para soldar
Ponto1, ponto 2, pontos 3 ...até que toda a placa seja soldada
E por fim o braço volta para o ponto inicial, para se preparar para a próxima placa
Tornando o percurso do braço um ciclo. Que pode recomeçar caso seja necessário
Como os robôs são dispositivos caros
Queremos que o tempo de realização da tarefa seja minimizado

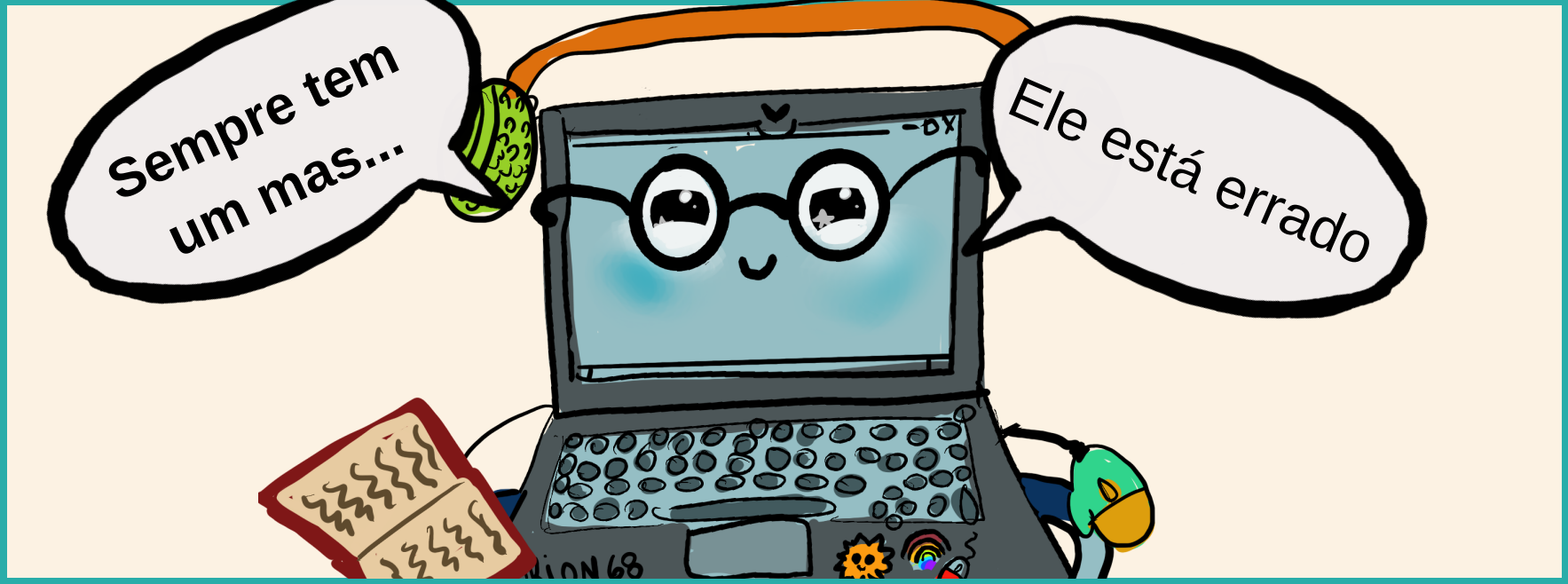
Problema: Otimização do Tour do Robô
Entrada: Um conjunto **S** de **n** pontos no plano.
Saída: Qual é o passeio de **<u>ciclo mais curto</u>** que visita cada ponto do conjunto

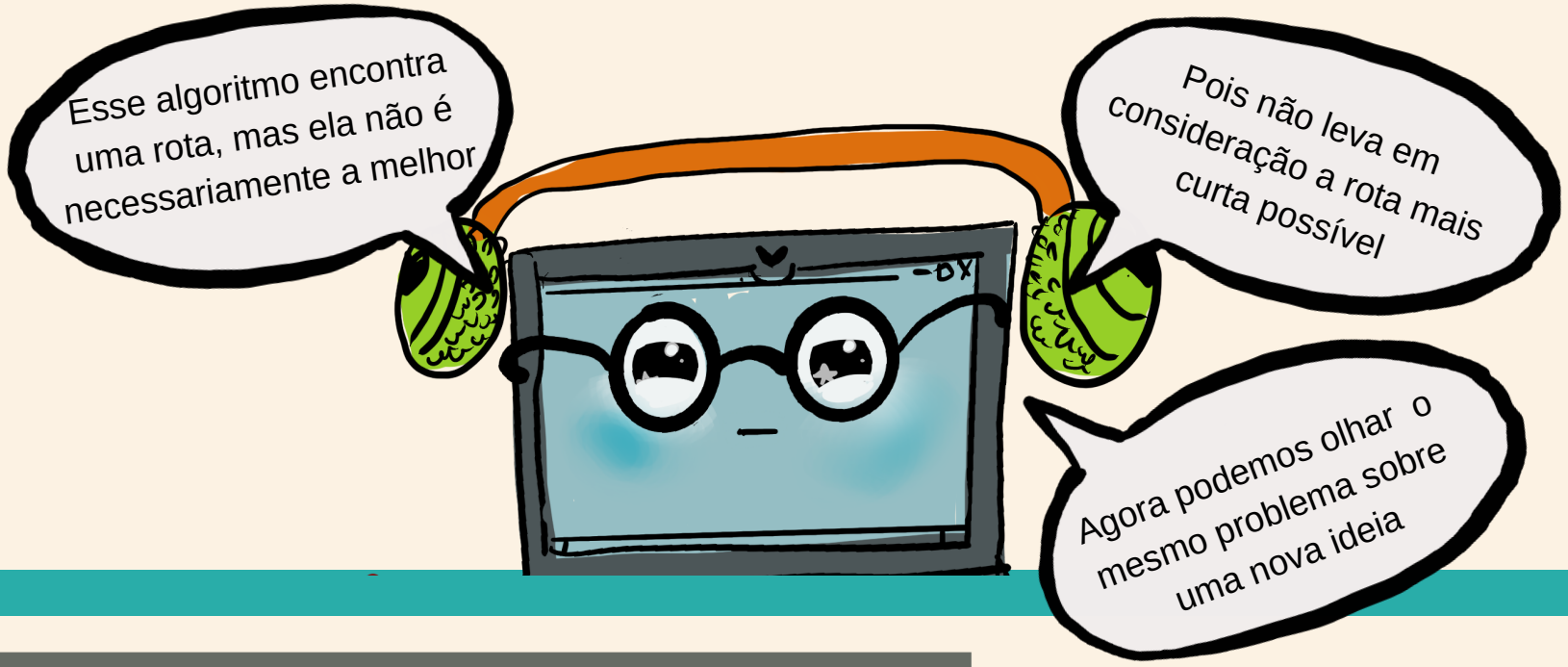
Vocês podem ter pensado em varias soluções para esse problema
Uma das ideias que pode ter passado na sua cabeça é a ideia do vizinho mais próximo
Partindo do ponto **P0**, caminhamos para o vizinho mais próximo **P1**, do **P1** caminhamos até o próximo vizinho não visitado, por fim voltamos para o **P0**
AGORA PENSEM
COMO VOU RESOLVER O PROBLEMA?
ORION 68

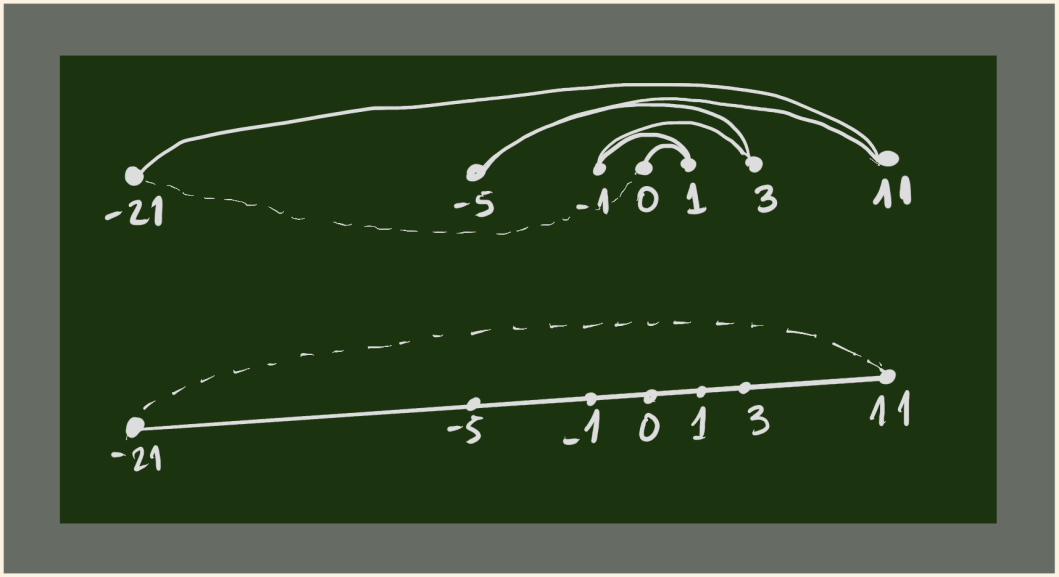

P0
P1
P2
P3
P4
P5
P6
P7

Fiz umas imagens para representar melhor a ideia
-DX
ORION 68


Esse algoritmo é simples e fácil de implementar, e funciona bem
MAS...

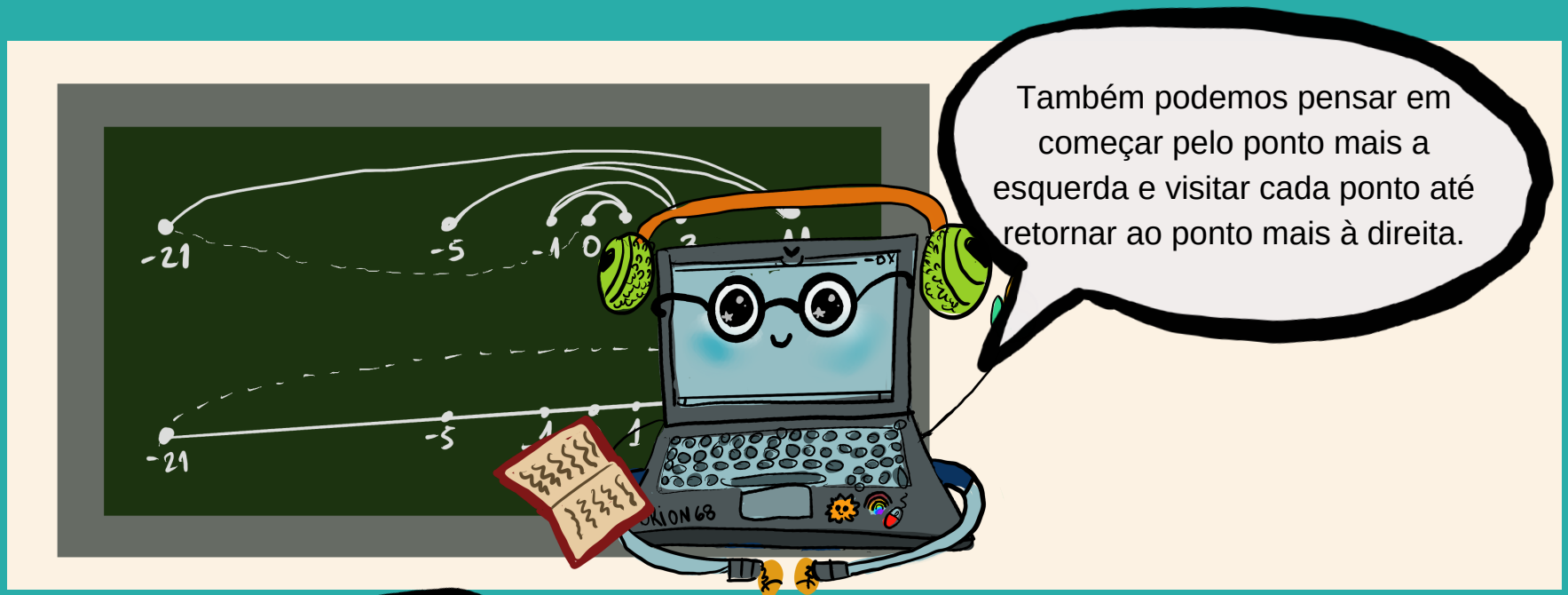

Sempre tem um mas...
Ele está errado

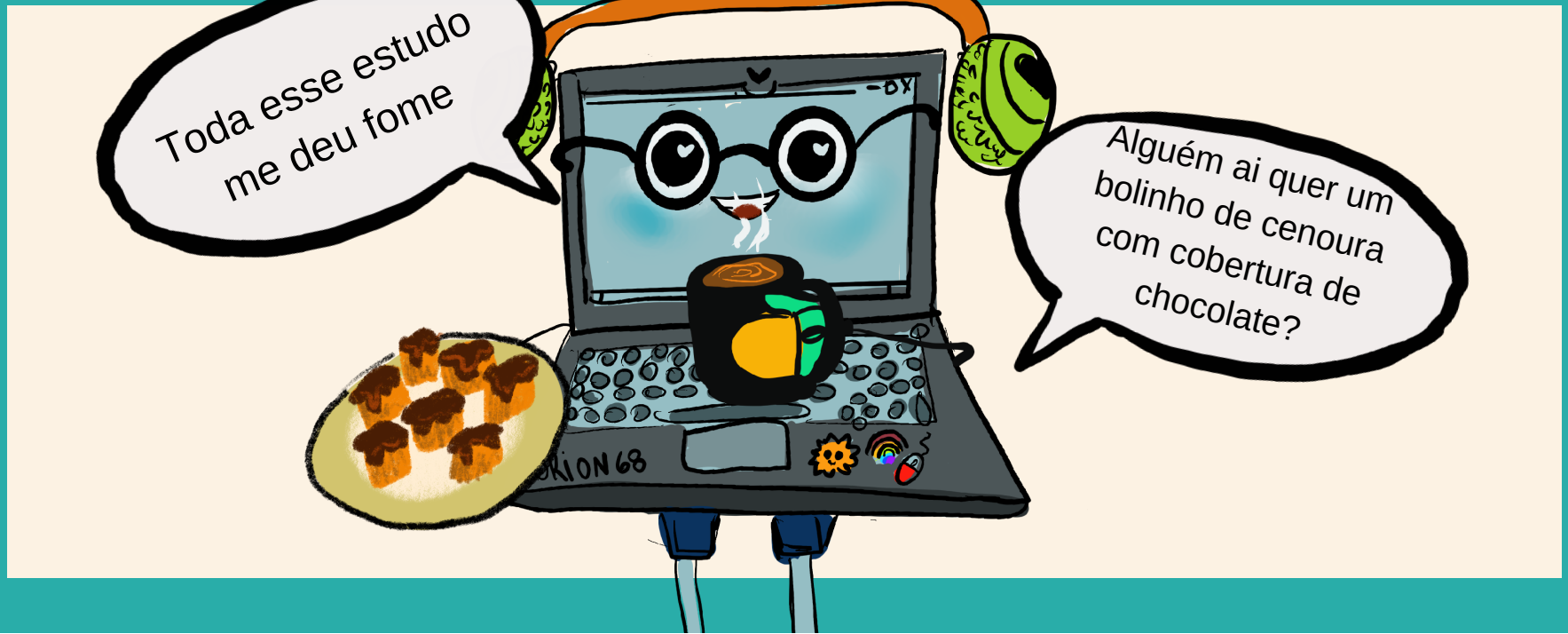


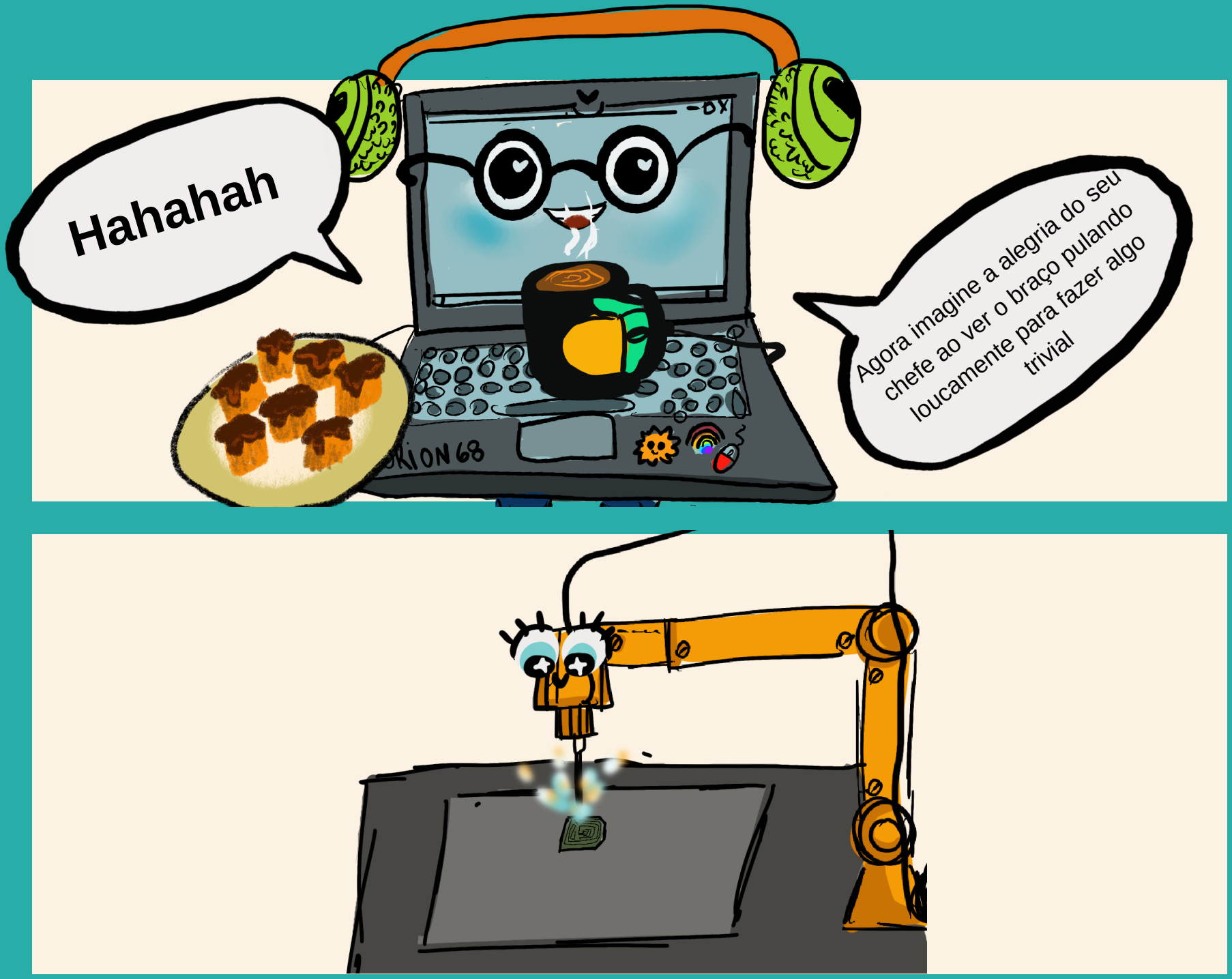


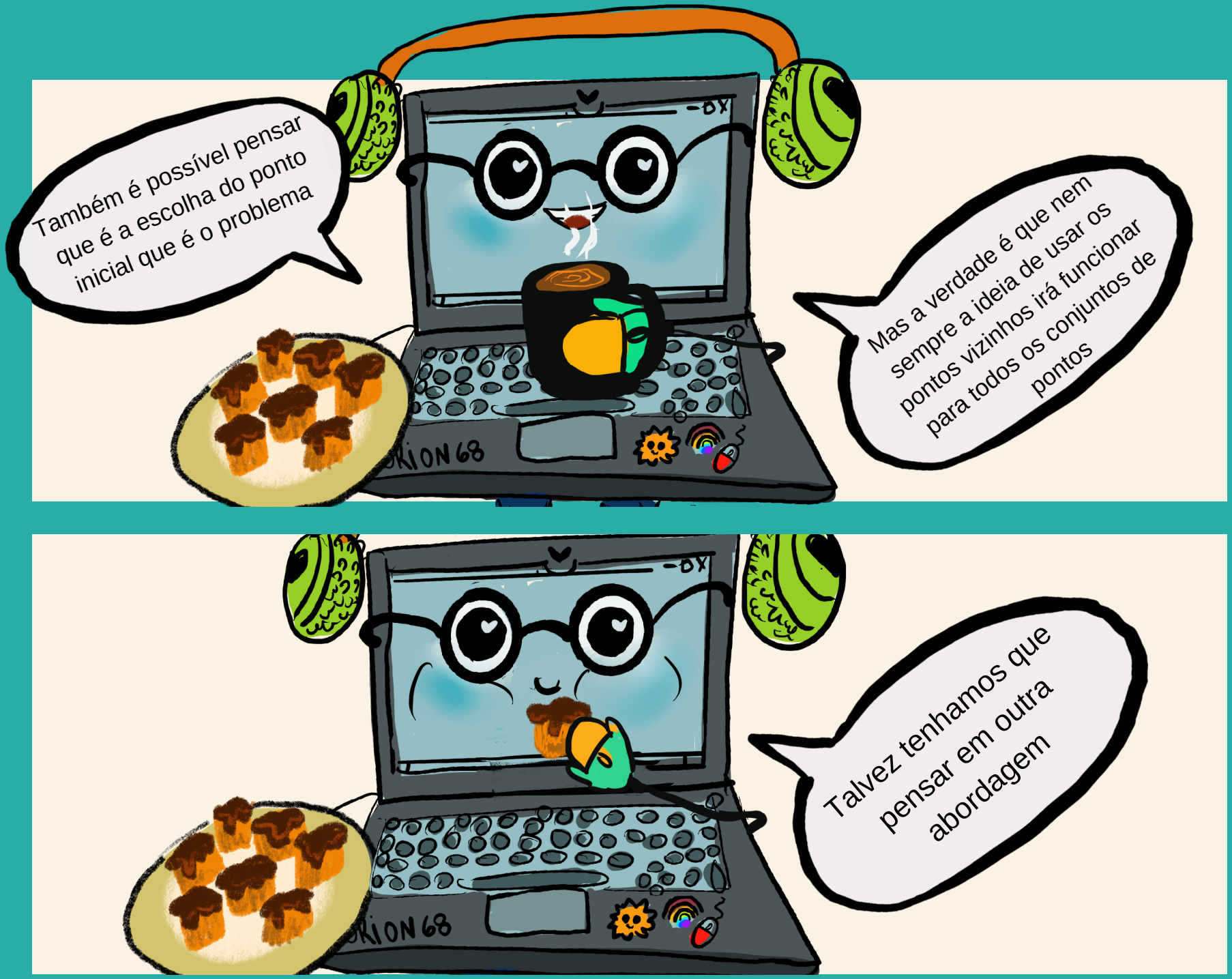
Imagine o números em uma reta os números mostram a distância que os pontos estão a esquerda ou à direita do ponto "0"

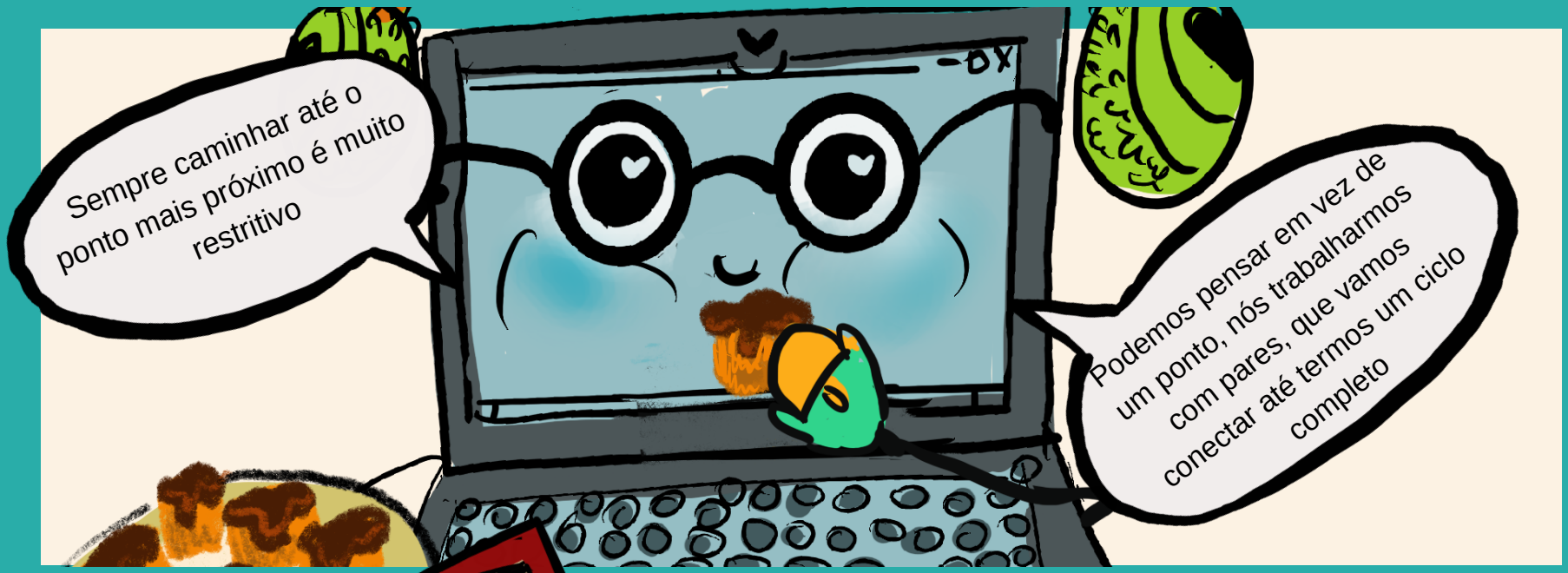
Quando começamos no ponto "0" e caminhamos repetidamente podemos pular entre os números, da esquerda para a direita, já que não há uma regra que nos direcione.


-21
-5
-1
0
-21
-5
Também podemos pensar em começar pelo ponto mais a esquerda e visitar cada ponto até retornar ao ponto mais à direita.

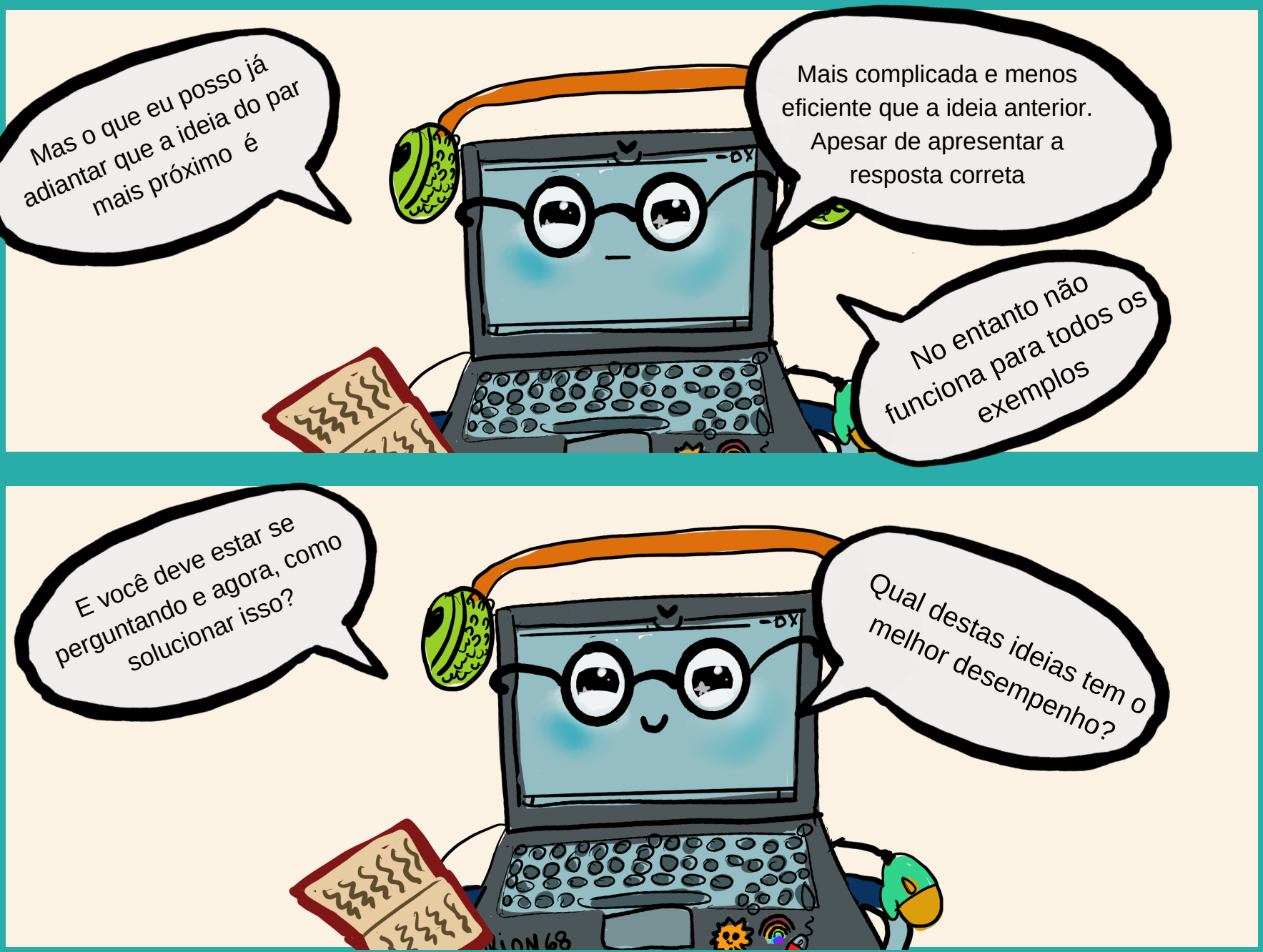

Toda esse estudo me deu fome
Alguém ai quer um bolinho de cenoura com cobertura de chocolate?

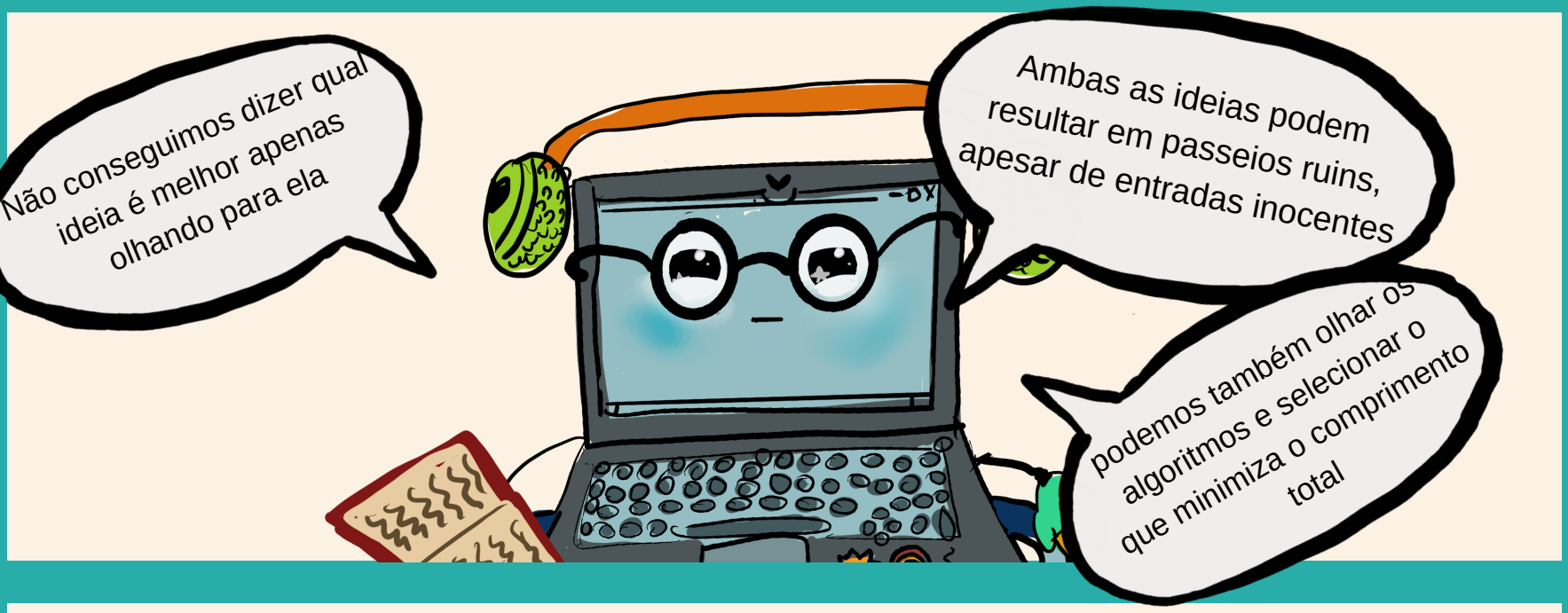

Hahahah
Agora imagine a alegria do seu chefe ao ver o braço pulando loucamente para fazer algo trivial
ORION 68


Também é possível pensar que é a escolha do ponto inicial que é o problema
Mas a verdade é que nem sempre a ideia de usar os pontos vizinhos irá funcionar para todos os conjuntos de pontos
Talvez tenhamos que pensar em outra abordagem

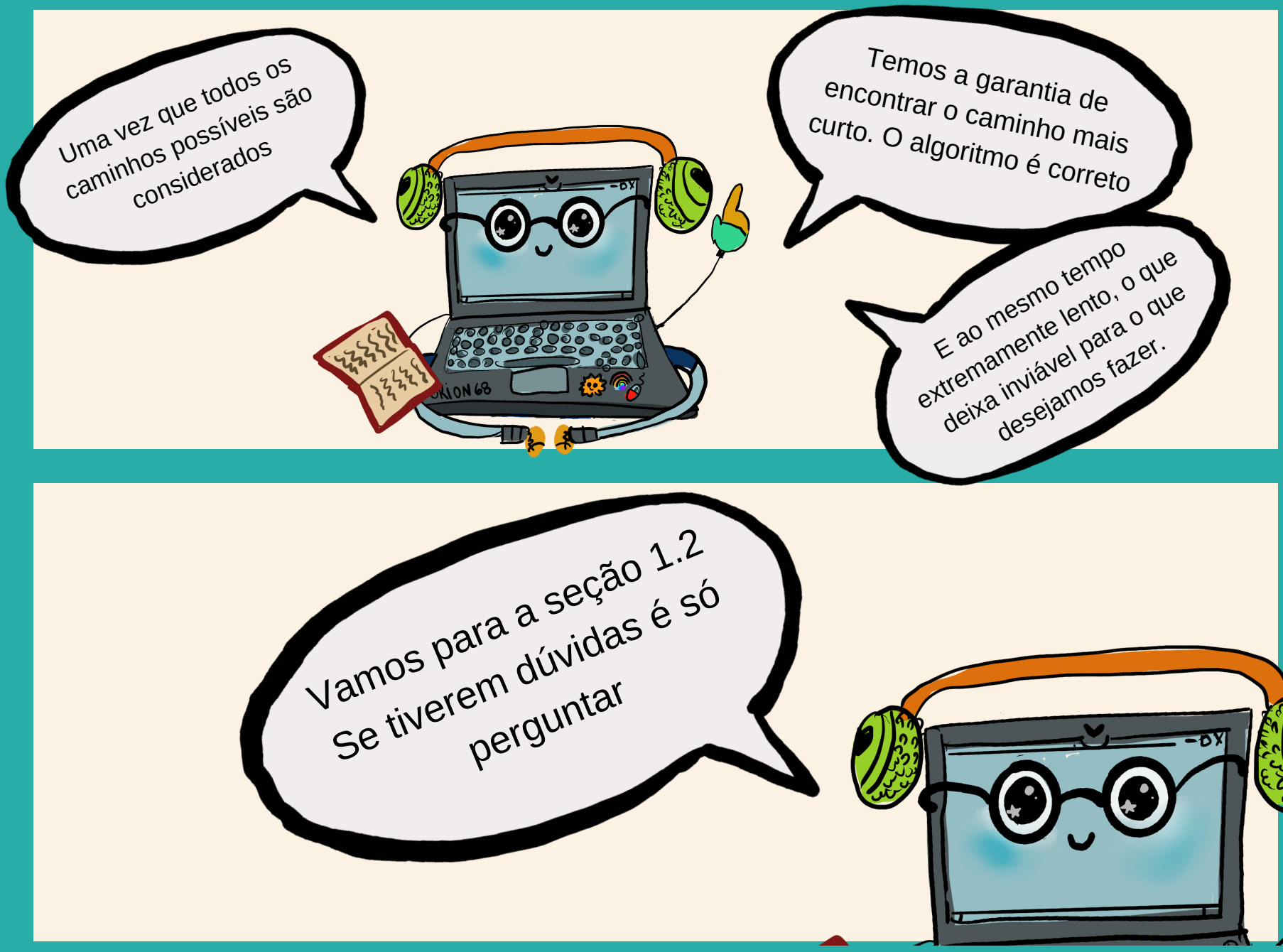


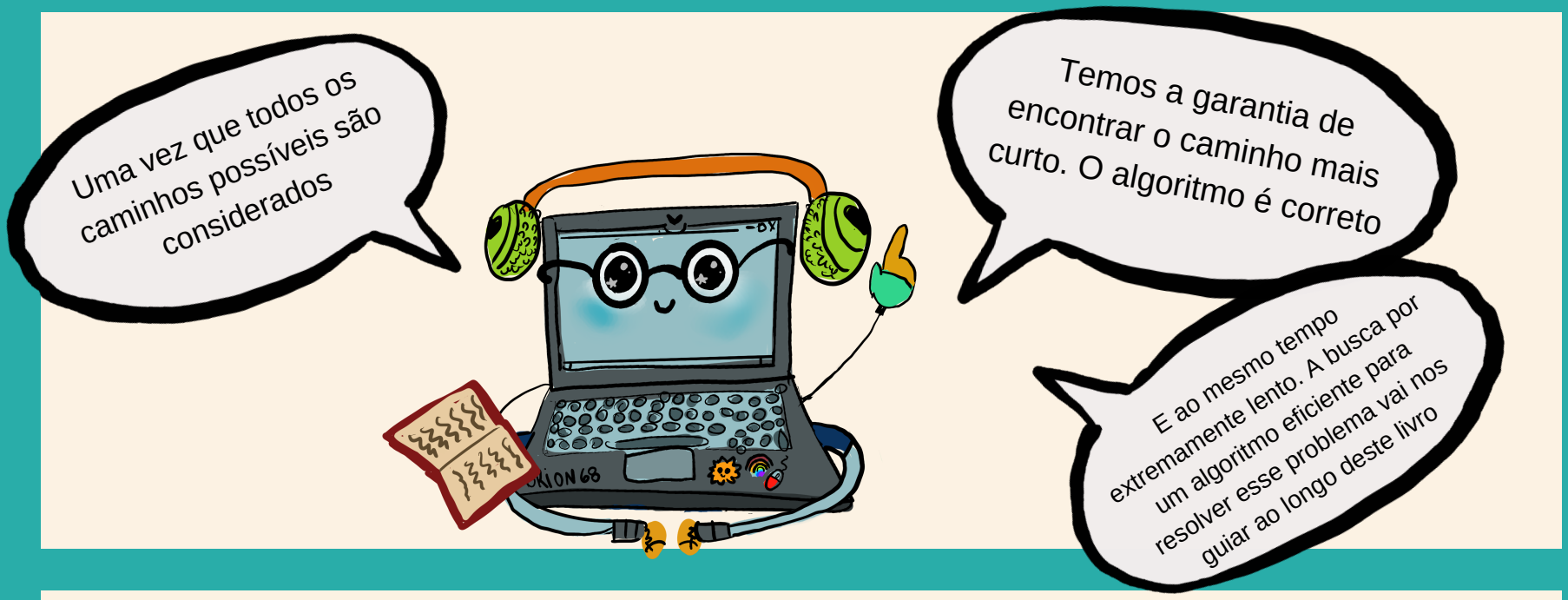
No livro podemos observar o pseudocódigo desta ideia

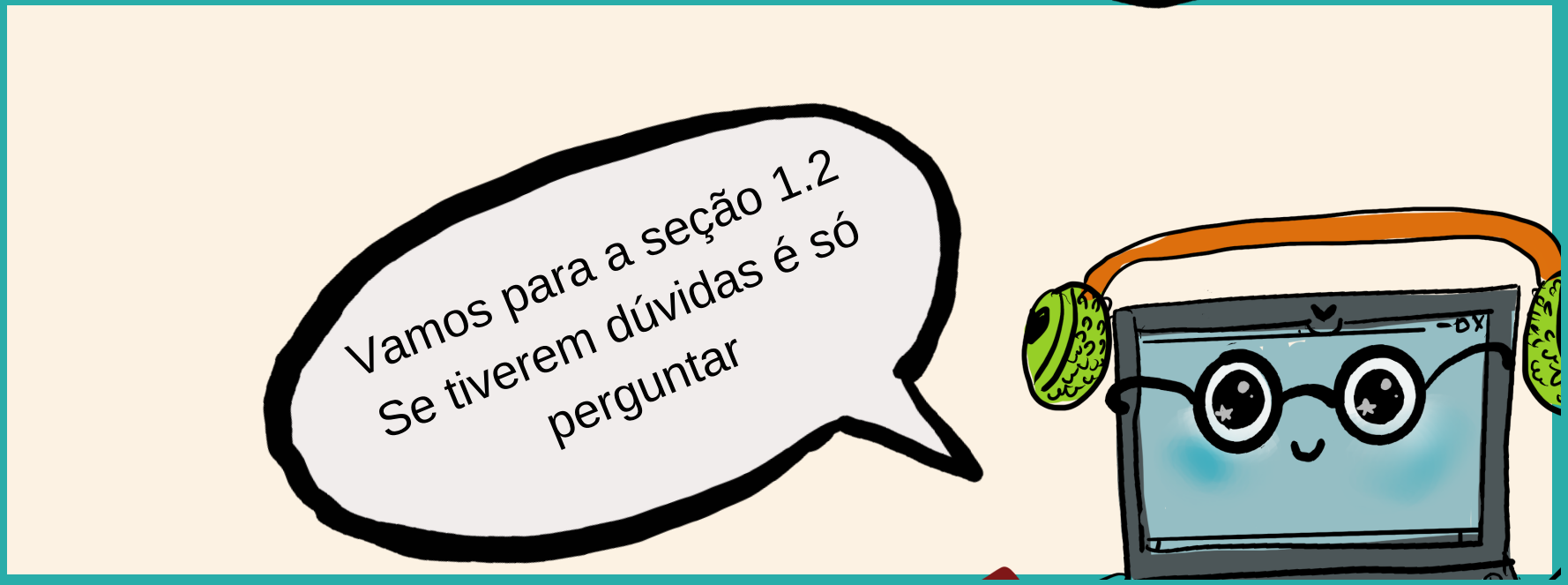

Mas o que eu posso já adiantar que a ideia do par mais próximo é
Mais complicada e menos eficiente que a ideia anterior. Apesar de apresentar a resposta correta
No entanto não funciona para todos os exemplos
E você deve estar se perguntando e agora, como solucionar isso?
Qual destas ideias tem o melhor desempenho?




```
TSP_Ideal(P)
    d = ∞
    Para cada um dos n! permutações P_i do conjunto de
pontos P
        Se (custo(P_i) ≤ d) então d = custo(P_i) e P_min = P_i
    Retornar P_min
```


Uma vez que todos os caminhos possíveis são considerados
Temos a garantia de encontrar o caminho mais curto. O algoritmo é correto
E ao mesmo tempo extremamente lento, o que deixa inviável para o que desejamos fazer.
Vamos para a seção 1.2 Se tiverem dúvidas é só perguntar


Uma vez que todos os caminhos possíveis são considerados
Temos a garantia de encontrar o caminho mais curto. O algoritmo é correto
E ao mesmo tempo extremamente lento. A busca por um algoritmo eficiente para resolver esse problema vai nos guiar ao longo deste livro


Vamos para a seção 1.2
Se tiverem dúvidas é só perguntar

Vamos imaginar que sou um ator muito requisitado
ORION 68
E que recebi várias propostas de projetos para fazer n Filmes
Todas as propostas vem com o primeiro e o último dia de filmagem

Agora preciso de ajuda para saber quais propostas devo aceitar
Lembrando que preciso estar disponível durante todo o período das gravações
E não posso aceitar trabalhos que tenham datas que se sobrepõem

Pensem a respeito, vou ter que fazer uma pausa agora
Na próxima aula terminamos de ler esse capítulo
ORION 68