

Engenharia da Computação – 3ª série

Classes de Problemas P e NP

(E1, E2)

2024

Pergunta

- O que é Algoritmo?



Resposta



- Um **algoritmo** é uma sequência não-ambígua de instruções para resolver um problema, *i.e.*, para obter uma saída requerida a partir de qualquer entrada legítima em uma quantidade finita de tempo.

Resposta



- Segundo *Cormen*, um **algoritmo** é qualquer procedimento computacional bem definido que toma algum valor, ou conjunto de valores, como entrada e produz algum valor, ou conjunto de valores, como saída;
- Portanto, um algoritmo é uma sequência de passos computacionais que transformam a entrada na saída;
- Ferramenta para resolver um problema computacional bem especificado.

Classes de Problemas P e NP

Resposta



- De uma maneira informal, um algoritmo é um método passo-a-passo para resolver um problema;
- O termo se aplica tipicamente para uma solução que pode ser executada em um computador.

Pergunta

- Por que estudar Algoritmos?



Resposta



- Há razões de ordem **Prática** e **Teórica** para se estudar um **algoritmo**;
- “Algoritmos representam mais que uma área da computação. Correspondem ao núcleo da Ciência da Computação, e com toda a imparcialidade, podem ser considerados relevantes para a maioria das ciências, negócios e tecnologia” – David Harel, 1992.

Pergunta

- Quais são as razões de Ordem Prática para se estudar Algoritmos?



Classes de Problemas P e NP

Resposta



- Há **algoritmos** em diversas áreas e precisa-se conhecê-los;
- Deve-se ter condições de avaliar algoritmos no que tange à sua eficiência;
- Deve-se ser capaz de projetar novos algoritmos.

Pergunta

- Quais são as razões de Ordem Teórica para se estudar Algoritmos?



Classes de Problemas P e NP

Resposta



- O estudo dos **algoritmos** é reconhecidamente considerado a base da Ciência da Computação.

Pergunta

- Qual é a origem do termo Algoritmo?



Classes de Problemas P e NP

Resposta



- A palavra **algoritmo** tem origem no sobrenome, *Al-Khwarizmi*, do matemático persa do século IX, *Mohamed ben Musa*;
- <http://pt.wikipedia.org/wiki/Algoritmo>

Pergunta

- O que é Especificação do Algoritmo?



Classes de Problemas P e NP

Resposta

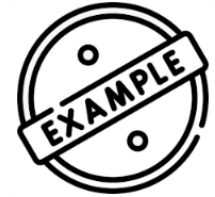


- O enunciado do problema especifica, em termos gerais, o relacionamento entre a entrada e a saída desejada.

ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Exemplo

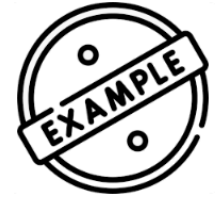


- Considerando-se um problema de ordenação (*sorting*) de uma sequência de números;
- Este problema surge na prática com frequência e oferece um “solo fértil” para o desenvolvimento de muitas técnicas de projeto e ferramentas de análise

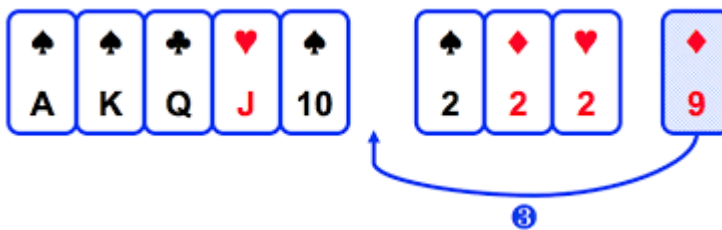
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Exemplo



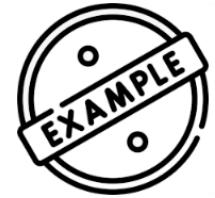
- E como deve ser especificado o *sorting*?
- Entrada: uma sequência de números: $[a_1, a_2, \dots, a_n]$;
- Saída: uma permutação (reordenação): $[a'_1, a'_2, \dots, a'_n]$ da sequência de entrada, tal que $[a'_1 \leq a'_2 \leq \dots \leq a'_n]$;



ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Exemplo



Sorting

problema

algoritmo



input

"computador"

output

$[a_1, a_2, \dots, a_n];$

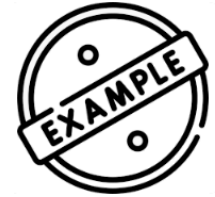
$[a'_1, a'_2, \dots, a'_n]$

$[a'_1 \leq a'_2 \leq \dots \leq a'_n];$

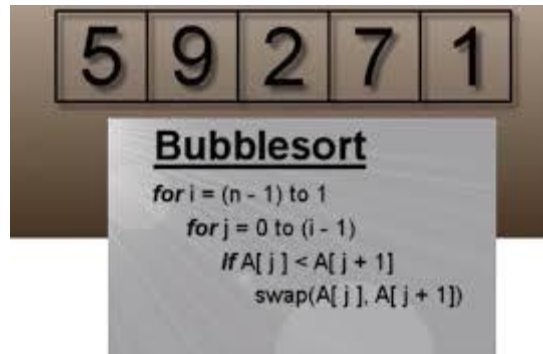
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Exemplo



- A ordenação é uma operação fundamental em Ciência da Computação;
- Muitos programas a utilizam como uma etapa intermediária;
- Como resultado, um grande número de bons algoritmos de ordenação têm sido desenvolvidos.



Pergunta

- Que tipo de problemas são resolvidos por Algoritmos?



ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- Biologia Computacional;
- Roteamento de tráfego Internet;
- Comércio Eletrônico: criptografia;
- Rotas rodoviárias.



*Ciência da
Computação*

Pergunta

- Quais problemas são estudados, então?



ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- Ordenação (*Sorting*);
- Busca (*Searching*);
- Processamento de *Strings*;
- Problemas geométricos;
- Problemas numéricos;
- Problemas combinatórios;
- Caixeiro Viajante;
- Jogos;
- Torres de Hanoi;
- Otimização.



Conclusão



- Os Algoritmos devem produzir saídas corretas!



Pergunta

- O que é Corretismo?



Resposta



- **Algoritmos corretos** usualmente são acompanhados por uma prova formal, o qual explica o porque o algoritmo gera saídas corretas para todas as instâncias do problema;
- Sem a prova matemática, as vezes, pode haver enganos a respeito do **corretismo** de um algoritmo;
- A intuição, muitas vezes, pode levar a resultados errôneos!

Pergunta

- Então, **Algoritmos Incorretos** devem ser **descartados**?



Classes de Problemas P e NP

Resposta



- Ao contrário do que se poderia esperar, muitas vezes, os **algoritmos incorretos** podem ser **úteis**, se sua taxa de erros puder ser controlada;
- Por exemplo, os algoritmos utilizados para se localizar grandes números primos!

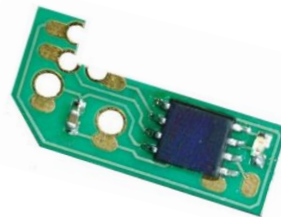
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Exemplo



- Para ilustrar o **Corretismo**, supõe-se um robô equipado com uma ferramenta de solda, para soldar pontos de um placa de circuito.
- O robô deve executar o trabalho de solda em determinados pontos de contato;
- O robô recebe uma quantidade de pontos de contato, devendo visitar o primeiro, o segundo, etc. .. até finalizar o trabalho.



ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Exemplo

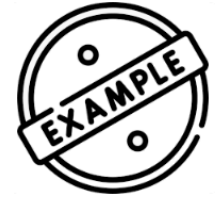


- Robôs são **caros**!
- Assim, quer-se **minimizar** o tempo que o braço do robô leva para soldar os pontos de contato;
- Assumindo-se, então, que o braço do robô se move com velocidade constante;
- Assim, o tempo para processar a placa é proporcional à distância percorrida pelo braço do robô...

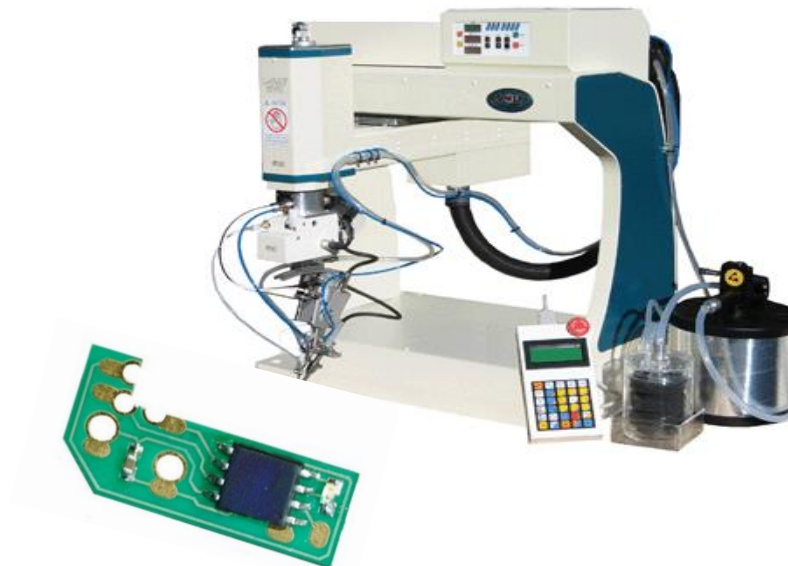
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Exemplo



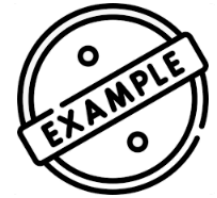
- **INPUT:** Um conjunto S de pontos num plano;
- **OUTPUT:** O caminho mais curto para visitar todos os pontos do ciclo.



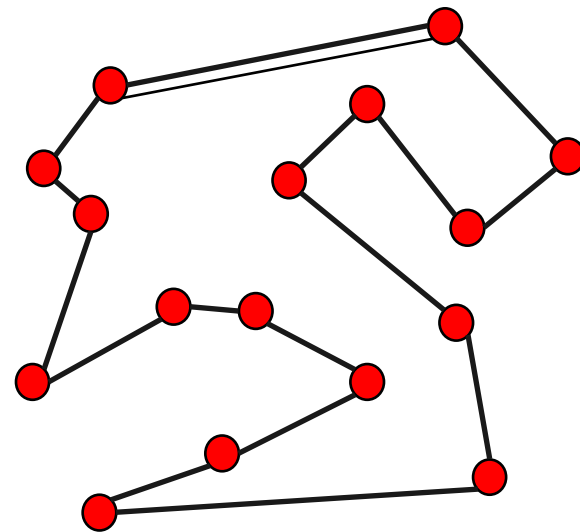
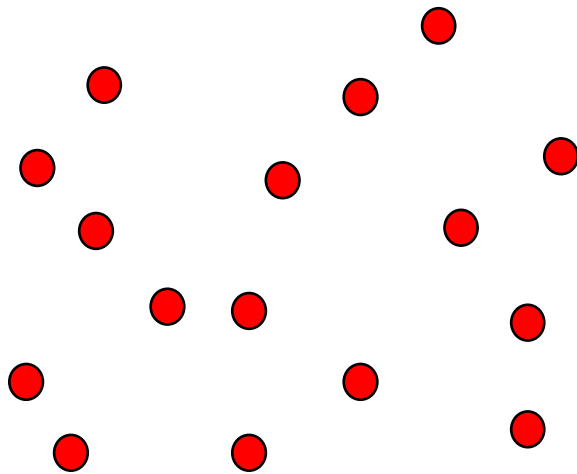
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Exemplo



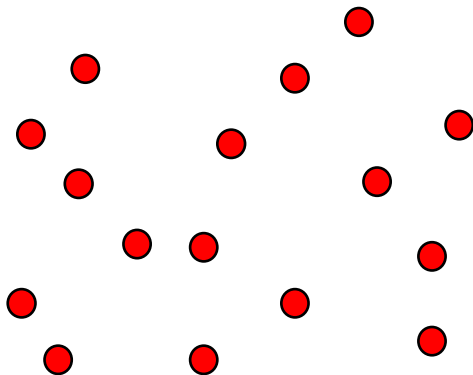
- Deve-se, então, encontrar o ciclo mais curto para visitar todos os pontos de solda, por exemplo:



Exemplo



- Imaginando-se que você foi contratado para programar o braço do robô, pense um pouco em algum algoritmo capaz de resolver esse problema!



ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Exemplo



- Imaginando-se que você foi contratado para programar o braço do robô, pense um pouco em algum algoritmo capaz de resolver esse problema!



Pergunta

- O que é a Heurística de ***Nearest-Neighbor***?



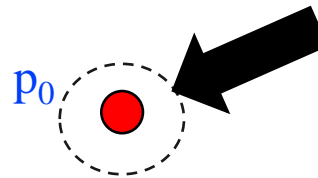
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- Inicia-se em algum ponto p_0 :



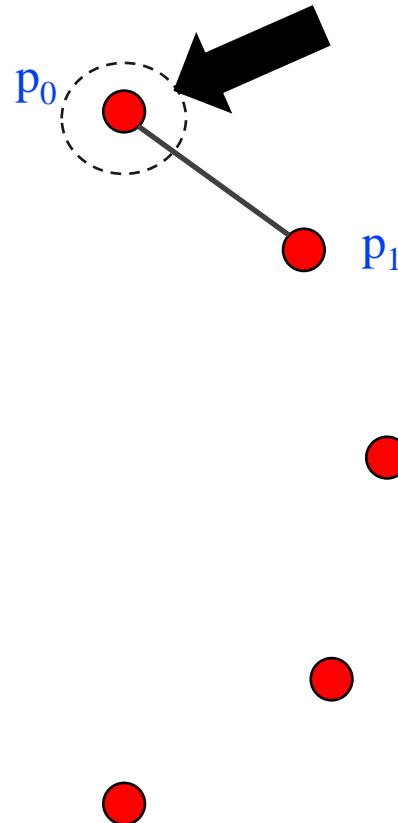
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- Do ponto p_0 prossegue-se até o seu vizinho mais próximo p_1 :



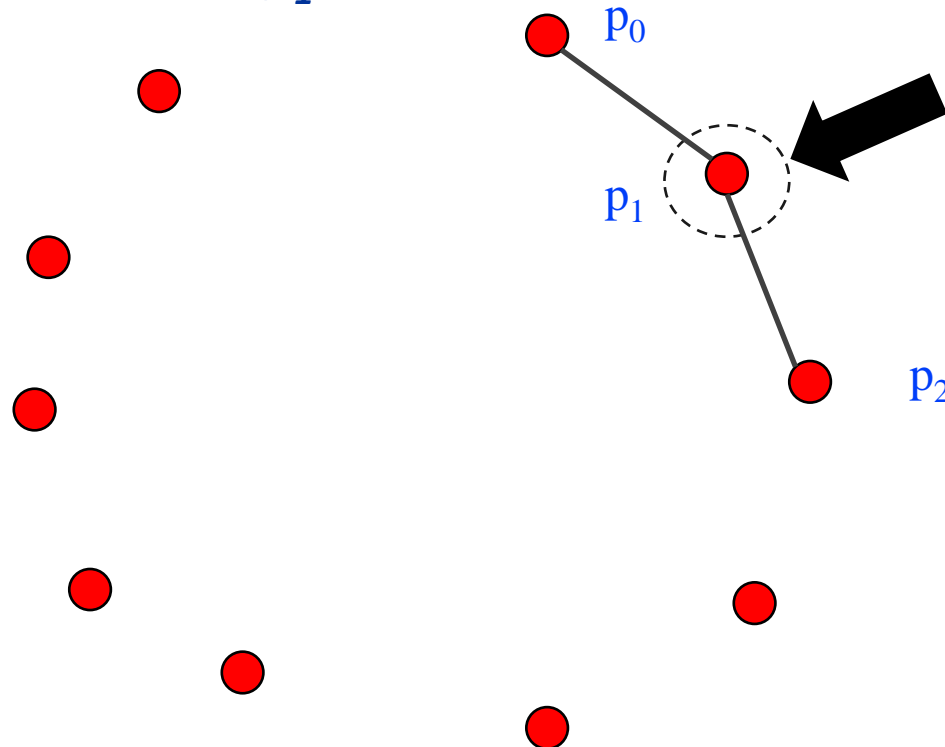
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- A partir de p_1 prossegue-se até o seu vizinho mais próximo não-visitado, excluindo-se p_1 :



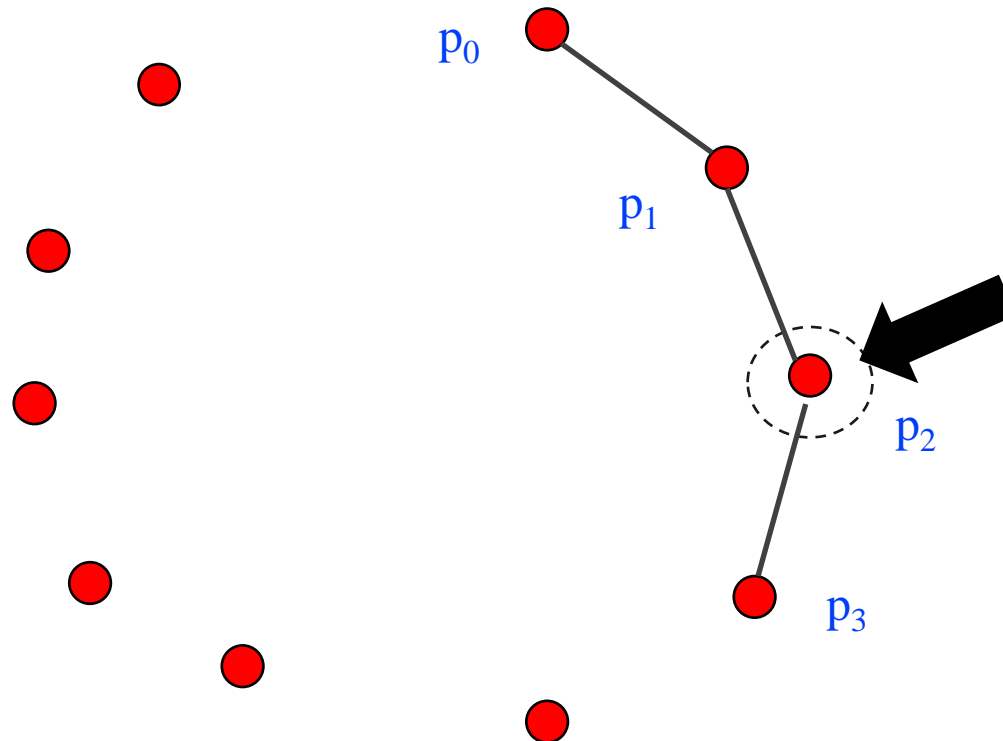
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- Repete-se o processo até passar por todos os pontos não-visitados:



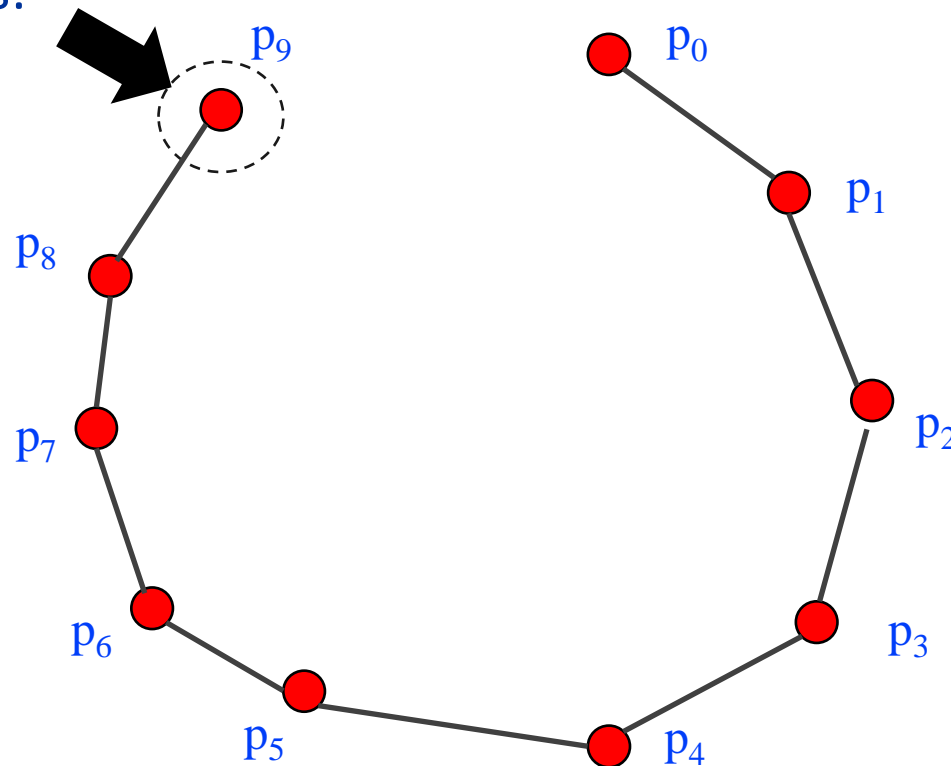
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



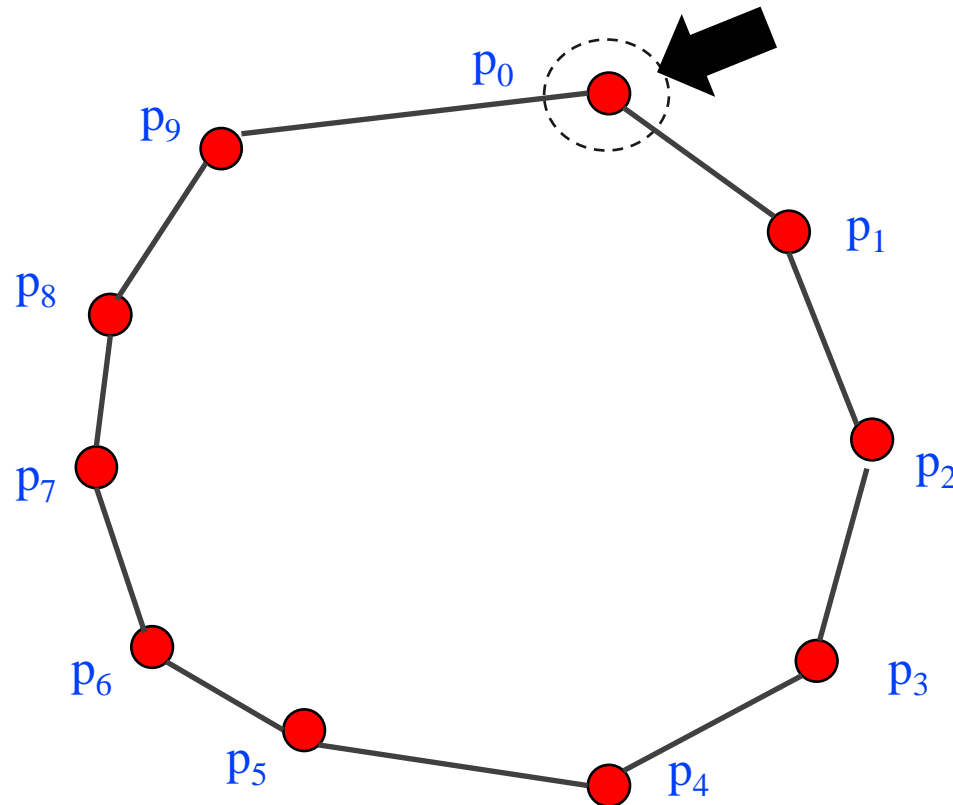
- Repete-se o processo até passar por todos os pontos não-visitados:



Resposta



- Ao final, retorna-se a p_0 , fechando-se o ciclo:



Classes de Problemas P e NP

Resposta



- Prazer! Esse é o algoritmo ***NearestNeighborTSP(P)***.
- O que você achou do algoritmo?



Resposta



- O algoritmo *NearestNeighborTSP(P)*:
- No entanto, esse algoritmo nem sempre retorna a melhor rota!



Resposta



- O algoritmo *NearestNeighborTSP(P)*:
- Considerando-se outra instância do problema, então!



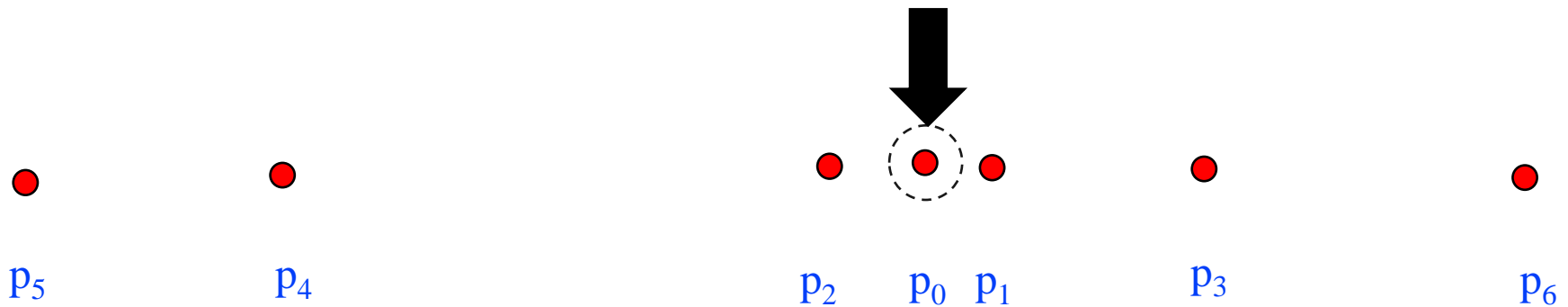
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- O algoritmo *NearestNeighborTSP(P)*:



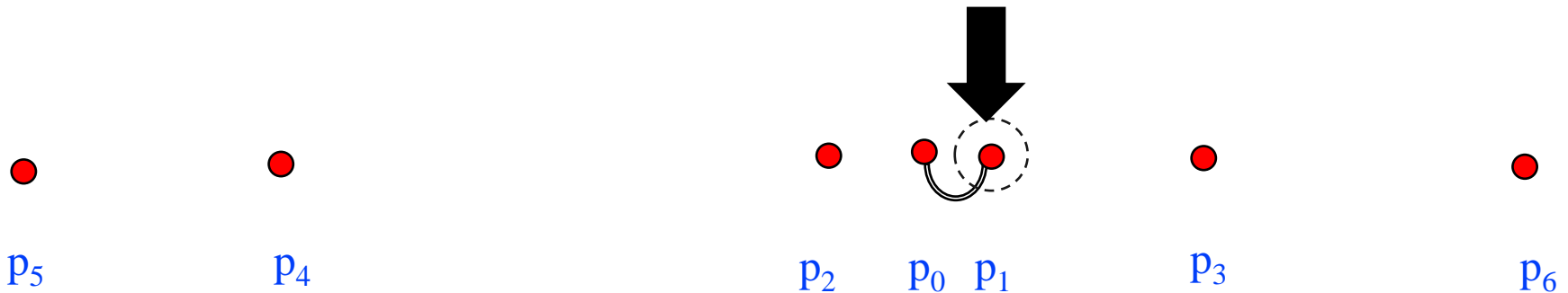
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- O algoritmo *NearestNeighborTSP(P)*:



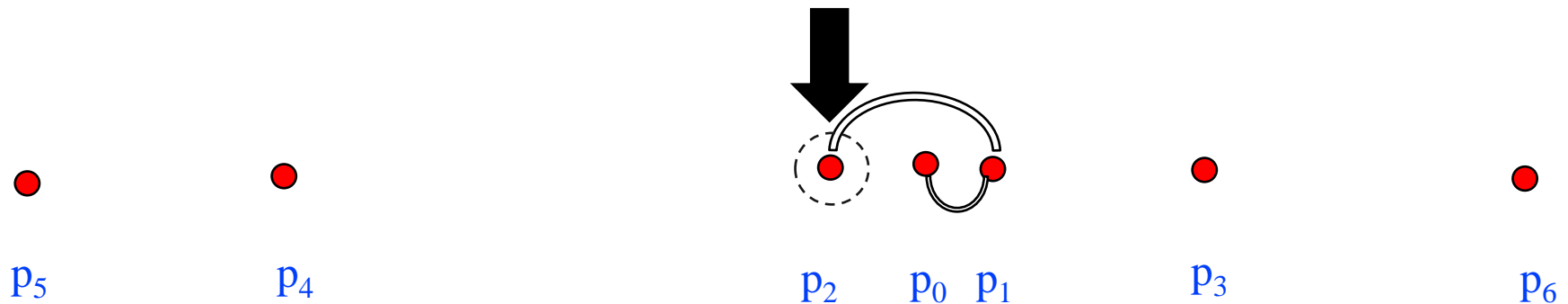
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- O algoritmo *NearestNeighborTSP(P)*:



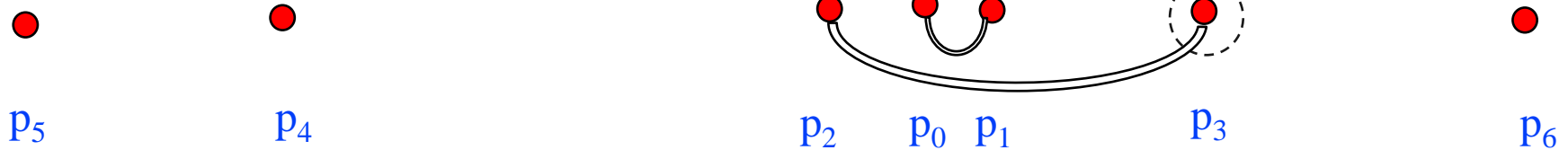
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- O algoritmo *NearestNeighborTSP(P)*:



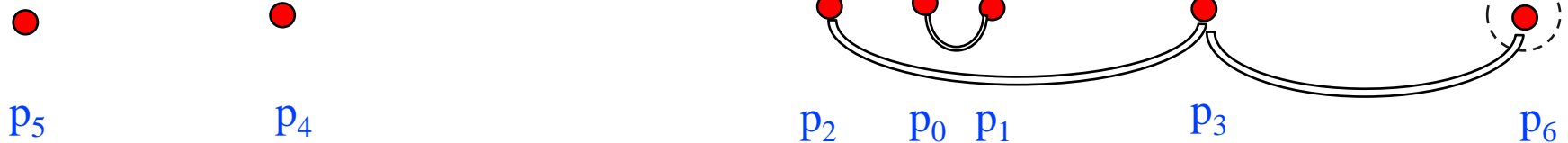
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- O algoritmo *NearestNeighborTSP(P)*:



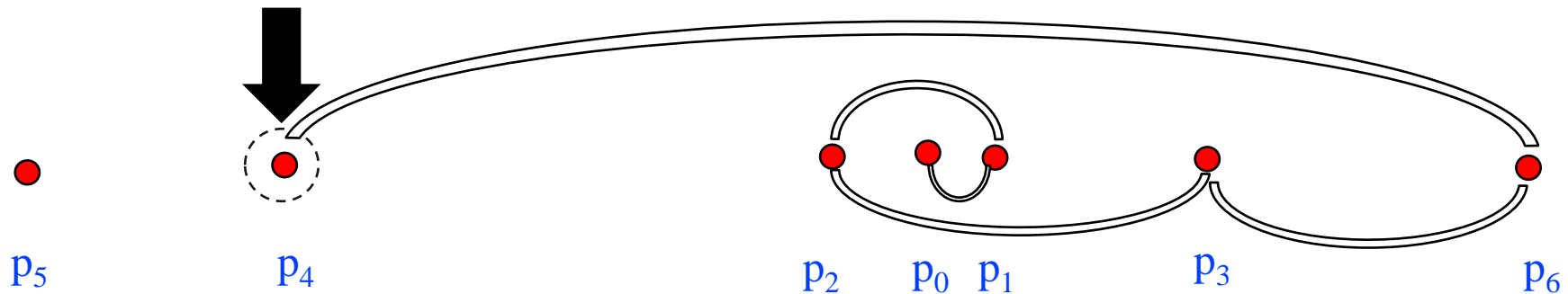
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- O algoritmo *NearestNeighborTSP(P)*:



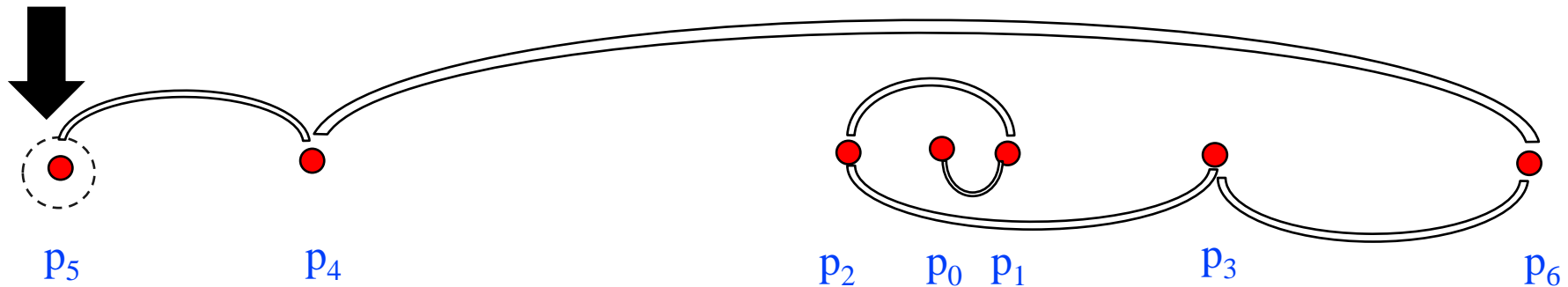
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



- O algoritmo *NearestNeighborTSP(P)*:



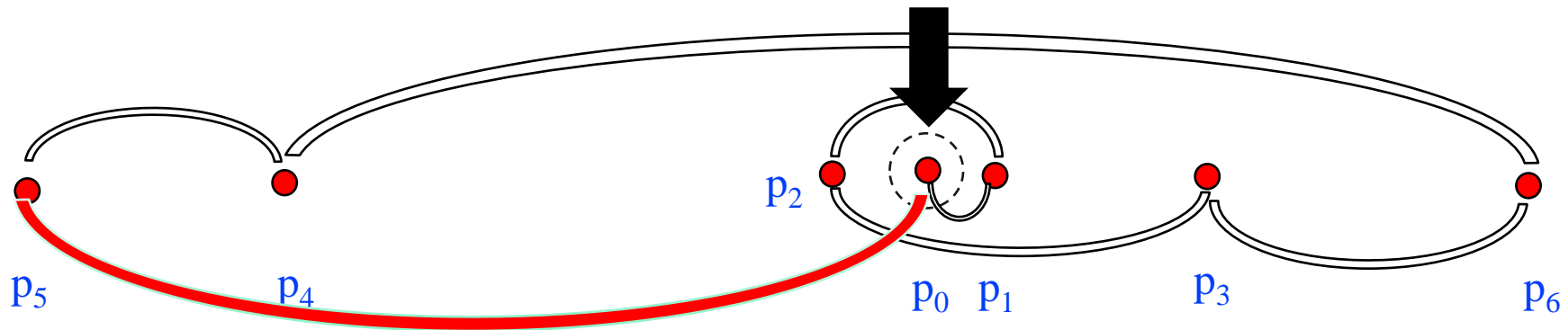
ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Resposta



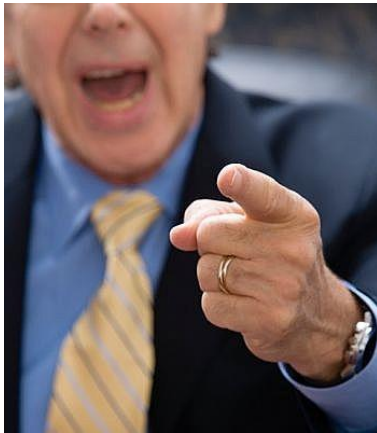
- O algoritmo *NearestNeighborTSP(P)*:



Resposta



- O algoritmo *NearestNeighborTSP(P)*:
- Imagine o seu chefe vendo esse vai-e-vem do robô...



Pergunta

- Há algum algoritmo melhor?



Resposta



- Sim, o algoritmo de **Busca Exaustiva**!
 - Tente todas as possíveis combinações de pontos;
 - Selecione dentre estas aquela que tem o menor custo.
- Desde que todas as combinações estão sendo consideradas, sempre, o algoritmo de **Busca Exaustiva** é o mais **correto** para essa situação!

Resposta



- Sim, o algoritmo de **Busca Exaustiva!**
 - Tente todas as possíveis combinações de pontos;
 - Selecione dentre estas aquela que tem o menor custo.
- Desde que todas as combinações estão sendo consideradas, sempre, o algoritmo de Busca Exaustiva é o mais correto para essa situação!
- **No entanto...**

Resposta



- Sim, o algoritmo de **Busca Exaustiva!**
 - Tente todas as possíveis combinações de pontos;
 - Selecione dentre estas aquela que tem o menor custo.
- Desde que todas as combinações estão sendo consideradas, sempre, o algoritmo de Busca Exaustiva é o mais correto para essa situação!
- No entanto, **o algoritmo é extremamente lento!**

ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Exemplo



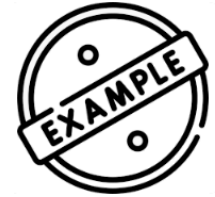
- Para 20 pontos:
 - Tem-se: $20!$ combinações
 - $20! = 2.432.902.008.176.640.000$ combinações
 - **Extremamente lento!!!**



ECM306 – Tópicos Avançados em Estrutura de Dados

Classes de Problemas P e NP

Exemplo



- Para circuitos reais, onde é comum $n \approx 1000$ pontos:
 - Tem-se: $1000!$ combinações
 - **Esqueça esse algoritmo!!!**



Conclusão



- O problema apresentado é clássico na Computação!
- Denomina-se TSP – *Travelling Salesman Problem*, ou, em Português, PCV – Problema do Caixeiro Viajante;
- **Não existe algoritmo correto e eficiente para este problema!**
- O problema é atacado com **Heurísticas**.



Pergunta

- O que são Algoritmos Heurísticos?



Classes de Problemas P e NP

Resposta



- Na Ciência da Computação, busca-se criar algoritmos com tempo de execução **aceitável** e ser uma solução **ótima** para o **problema** em **todas** as suas **instâncias**;
- Um **algoritmo heurístico** não cumpre uma dessas propriedades, podendo ser um algoritmo que encontra boas soluções a maioria das vezes, mas não há garantias de que sempre as encontrará.

Pergunta

- O que são Algoritmos Polinomiais?



Resposta



- A **Complexidade Polinomial** representa o divisor de águas entre as Classes de Algoritmos;
- **Algoritmos Polinomiais** são considerados **tratáveis**;
- **Algoritmos** com **complexidades superiores** às **polinomiais** são considerados **intratáveis**, por exemplo, o algoritmo do Caixeiro Viajante, ou TSP – *Travelling Salesman Problem*.

Classes de Problemas P e NP

Pergunta

- O que são as Classes \mathcal{P} ?



Classes de Problemas P e NP

Resposta



- O estudo da teoria da complexidade de algoritmos concentra-se nos **problemas de decisão**;
- Definição:

A **Classe \mathcal{P}** representa o conjunto dos **problemas de decisão** que **podem** ser resolvidos por um **Algoritmo Polinomial**.

Classes de Problemas P e NP

Pergunta

- O que são as Classes NP ?



Classes de Problemas P e NP

Resposta



- Definição:

A **Classe NP** representa o conjunto dos **problemas de decisão** em que dada qualquer instância do problema para a qual a resposta é “sim”, existe um certificado validando este fato e que pode ser verificado em tempo polinomial.

ECM306 – Tópicos Avançados em Estrutura de Dados

Referências bibliográficas



- CORMEN, T.H. et al. Algoritmos: Teoria e Prática (Caps. 13). Campus. 2002.
- ZIVIANI, N. Projeto de algoritmos: com implementações em Pascal e C (Cap. 1). 2.ed. Thomson, 2004.
- FEOFILOFF, P. Minicurso de Análise de Algoritmos, 2010. Disponível em:
<http://www.ime.usp.br/~pf/livrinho-AA/>
- DOWNEY, A.B. *Analysis of algorithms* (Cap. 2), Em: *Computational Modeling and Complexity Science*. Disponível em:
<http://www.greenteapress.com/compmod/html/book003.html>
- ROSA, J.L. Notas de Aula de Introdução a Ciência de Computação II. Universidade de São Paulo. Disponível em:
<http://coteia.icmc.usp.br/mostra.php?ident=639>

ECM306 – Tópicos Avançados em Estrutura de Dados

Referências bibliográficas

- GOODRICH, Michael T. et al: *Algorithm Design and Applications*. Wiley, 2015.
- LEVITIN, Anany. *Introduction to the Design and Analysis of Algorithms*. Pearson, 2012.
- SKIENA, Steven S. *The Algorithm Design Manual*. Springer, 2008.
- Série de Livros Didáticos. *Complexidade de Algoritmos*. UFRGS.
- BHASIN, Harsh. *Algorithms – Design and Analysis*. Oxford University Press, 2015.
- FREITAS, Aparecido V. de – 2022 – Estruturas de Dados: Notas de Aula.
- CALVETTI, Robson - 2015 – Estruturas de Dados: Notas de Aula.



ECM306 – Tópicos Avançados em Estrutura de Dados

Aula 09

FIM