



**unioeste**

Universidade Estadual do Oeste do Paraná

Campus de Foz do Iguaçu

CENTRO DE ENGENHARIAS E CIÊNCIAS EXATAS  
CURSO DE CIÊNCIA DA COMPUTAÇÃO



**PARANÁ**  
GOVERNO DO ESTADO

**3ª PROVA DE  
PROJETO E ANÁLISE DE ALGORITMOS**

Professora: Teresinha Arnauts Hachisuca

Data: 12/11/2014

Acadêmico: Gabriel Custódio Martins

- 28 1. (20) Sobre sistemas de computação paralela e distribuída. Pede-se:
- ☒ a. (10) O que são?
  - ☒ b. (5) Como são classificados?
  - ☒ c. (5) Quais os principais fatores que devem ser considerados para medir a eficiência dos mesmos?
- 15 2. (30) Sobre algoritmos de compressão de dados. Pede-se:
- ☒ a. (5) O que são?
  - ☒ b. (5) Como medir a eficiência dos mesmos.
  - ☒ c. (5) Quais os tipos compressão de dados? Descreva-os.
  - ☒ d. (15) Em relação ao trabalho feito pelo grupo, fale sucintamente dos detalhes de implementação e funcionamento.  
OK
- 49 3. (50) Com relação às classes de problemas. Fale sobre:
- ☒ a. (5) A Classe P.
  - ☒ b. (5) A Classe NP
  - ☒ c. (15) A questão  $P=NP$  ou  $P \neq NP$  ou  $P \subseteq NP$ ?
  - ☒ d. (5) A Classe NP-completo.
  - ☒ e. (5) A Classe NP-difícil.
  - ☒ f. (5) Problemas tratáveis e intratáveis.
  - ☒ g. (5) Quando um algoritmo é considerado eficiente?
  - ☒ h. (5) Como provar que um algoritmo pertence à classe NP ao invés de P?

1. a) Computação paralela e distribuída consiste de um conjunto de componentes de processamento (processadores) interconectados que buscam, de forma conjunta, solucionar um problema computacional.

b) São classificados através da taxionomia de Flynn de acordo com o fluxo de instruções e dados. A arquitetura de Von Neumann possui um fluxo único de dados e instruções, sendo desta forma classificado como SISD (single-instruction single data). Um conjunto de processadores independentes que executam paralelamente estão na categoria MIMD (multiple-instruction multiple-data). Intermediariamente a essas classes temos:

- SIMD - single-instruction multiple data
- MISD - multiple-instruction single-data

c) • Concorrência computacional

• Escalonamento

• Comunicação

• Sincronização

2. a) Buscam reduzir o tamanho do arquivo original eliminando redundância de dados, para realizar o armazenamento ou envio através da rede, possibilitando ou não a reconstrução do arquivo em seu formato original do lado do receptor.

b) Utilizando índices de taxa de compressão, consumo de processamento, tempo e memória do algoritmo e, caso houver, taxa de perda da qualidade.

c) Compressão de dados sem perda: Permite reduzir o tamanho do arquivo original, mas porém permite recuperá-lo sem que haja perda de informações.

Compressão de dados com perda: Reduz o tamanho do arquivo eliminando informações "desnecessárias", porém, não permite voltar ao estado original do arquivo.

d) O algoritmo de compactação utiliza uma variação do algoritmo de LZ com o uso de dicionário.

Dado um arquivo de texto de entrada, este é "quebrado" em palavras ou tokens que são agrupados, usando um algoritmo de hash, de forma a calcular a ocorrência de cada token e fornecendo um índice a cada um, que corresponde ao índice de sua primeira ocorrência no arquivo, caso este token ocorra mais de uma vez será armazenado em uma estrutura de dicionário e suas ocorrências no arquivo serão substituídas por seu índice. Desta forma tokens que ocorrem apenas uma vez são transcritos diretamente no arquivo.

Desta forma a saída consiste em um texto compactado juntamente com seu dicionário de dados.

3. a) A classe P consiste dos problemas de decisão aos quais existe um algoritmo polinomial que o soluciona em tempo razoável.  $\text{+}$

b) A classe NP consiste de problemas considerados "razáveis" onde dado um problema de decisão, proposto por um algoritmo não-determinístico, e seus respectivos dados  $\pi$ , é possível, através de um algoritmo em tempo polinomial, dizer se, utilizando os dados de entrada  $\pi$  e a solução encontrada, se esta está correta.  $\text{C}$

c) Apesar de algoritmos NP, consistirem daqueles aos quais não possui um algoritmo polinomial para resolver o seu problema, não pode-se assumir que este algoritmo possa um dia ser encontrado, além disso pode-se assumir que todos os problemas NP possuam uma solução polinomial, apenas não foi encontrada, desta forma pode-se dizer que  $P = NP$ , ou não, não há uma conclusão para este tema. Porém pode-se dizer que  $P \subseteq NP$ , já que um problema da classe P pode ser solucionado obedecendo aos critérios da classe NP.  $\text{C}$

d) Devem satisfazer as seguintes regras:

I. Ser NP

II. Dado um algoritmo, já conhecido B, NP-completo em que o algoritmo que se deseja atestar, A, seja uma especificação do problema B, e este problema A possa ser convertido para sua forma B a fim de ser verificado através do alg. NP completo e este processo for realizado com sucesso A é dito NP-completo

e) NP-difícil são problemas que precisam satisfazer apenas a segunda regra de NP-completo, desta forma NP-completo é a interseção de problemas NP e NP-difícil.

f) Problemas tratáveis são aqueles que podem ser solucionados em tempo razoável utilizando algoritmos polinomiais.

Problemas intratáveis são aqueles com soluções de algoritmos exponenciais, <sup>ou</sup> que de acordo com a quantidade de dados de entrada podem levar anos para finalizar.

g) Quando este soluciona um problema em tempo polinomial.  $\text{C}$

h) É considerado muito difícil, sendo que é necessário provar que todas as soluções de decisão possíveis e as que ainda serão criadas não possuam tempo polinomial.