



Arrays Associativos

Uma estrutura de dados poderosa que transforma a forma como resolvemos problemas computacionais complexos

Giovanni Milan Câmara Pinto, João Henrique Alves Silva, Samuel Alves Gomes Palaoro - Universidade Vila Velha

O Que São Arrays Associativos?

Estrutura de dados que permite indexação por chaves arbitrárias, como strings, ao invés de índices numéricos tradicionais.

Na linguagem AWK, todos os arrays são associativos por padrão, dispensando declarações explícitas e inicializações manuais.



Linguagem AWK



Processamento de Texto

Ferramenta poderosa para processamento de texto em sistemas UNIX/Linux



Busca de Padrões

Excelente para busca de padrões e geração de relatórios



Arrays Associativos

Suporte nativo a arrays associativos como estrutura central

Aplicações Clássicas



Contagem de Palavras

Cada palavra utilizada como índice do array para contagem de ocorrências em texto



Remoção de Duplicatas

Verificação de existência prévia de chave para identificar e remover duplicatas



Agregação de Dados

Mapeamento eficiente entre chaves e valores agregados



Verificação de Pertencimento

Busca rápida e direta para verificar existência de elementos

Contagem de Palavras

Implementação em AWK

Palavras são utilizadas diretamente como índices do array, simplificando drasticamente a contagem de ocorrências.

Elimina a necessidade de estruturas auxiliares complexas como listas encadeadas ou tabelas hash explícitas.



Aplicações Avançadas



Máquinas de Estados Finitos

Estados e transições representados por estruturas que mapeiam entradas e saídas



Ordenação Topológica

Algoritmo para organizar vértices de grafo direcionado de forma linear

Ordenação Topológica



Como Funciona

- Arrays associativos armazenam número de predecessores de cada nó
- Listas de sucessores mapeadas por chaves
- Filas controlam o processamento
- Solução eficiente com implementação simplificada

Comparação de Abordagens

Abordagens Tradicionais

Listas encadeadas, tabelas hash explícitas, estruturas mais complexas aumentam a complexidade do código

Arrays Associativos (AWK)

Representação direta de dados e relações, menor complexidade de implementação, código reduzido

Princípios Fundamentais

01

Escolha Representações Adequadas

A estrutura de dados adequada simplifica significativamente a resolução de problemas

02

Priorize a Simplicidade

Código claro e conciso antes de otimizações complexas

03

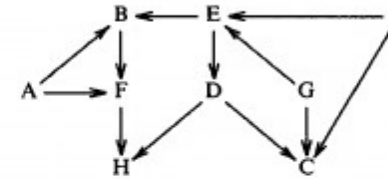
Desenvolva Soluções Claras

Clareza e facilidade de desenvolvimento são tão importantes quanto desempenho

Exemplos visuais:

NODE	PREDECESSOR COUNT	SUCCESSORS
A	0	B C
B	1	D
C	1	D
D	2	

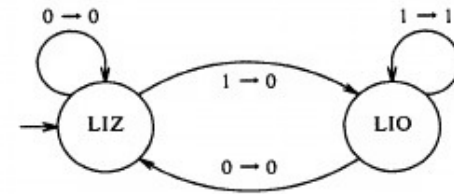
The input to a topological sorting algorithm is a directed graph with no cycles, such as



If the graph contains an edge from A to B , then A is B 's predecessor and B is A 's successor. The algorithm must order the nodes such that all predecessors appear before their successors; here is one of many possible orderings.



```
length($1) > 10 { e++; print "long name in line", NR}
NF != 1         { e++; print "bad name count in line", NR}
END             { if (e > 0) print "total errors: ", e }
```





Conclusão

A eficiência de um programa não está apenas em seu desempenho, mas também na clareza, concisão e facilidade de desenvolvimento.

Clareza do Código

Arrays associativos permitem soluções elegantes com código reduzido e fácil compreensão

Redução de Complexidade

Elimina necessidade de estruturas auxiliares elaboradas

Prototipação Rápida

Simplicidade torna adequada para desenvolvimento rápido