



Árvores múltiplas

Aleardo Manacero Jr.





Árvores Múltiplas



- Árvores múltiplas são formadas por nós que possuem mais do que dois filhos e, consequentemente, mais do que uma chave associada
- As chaves associadas a cada filho implicam em relações de ordem da mesma maneira que em árvores binárias
- Um exemplo representativo de árvores múltiplas é composto pelas árvores B



Árvores B



- Uma árvore B é uma forma de construção de árvores de busca em dispositivos de armazenamento secundário (discos), em que o custo de busca é reduzido pelo empacotamento de nós da árvore
- A idéia é fazer com que cada nó da árvore contenha tanta informação quanto a que cabe num bloco de disco



Árvores B



- Isso implica nas seguintes restrições para uma árvore de ordem m :
 - A raiz tem pelo menos duas sub-árvores
 - Cada nó intermediário tem $k-1$ chaves e k sub-árvores, em que $m/2 \leq k \leq m$
 - Cada nó folha tem $k-1$ chaves
 - Todas as folhas estão no mesmo nível



Inserção em árvores B



- Os elementos são inseridos sempre nas folhas, desde que se mantenha a profundidade da árvore
- Isso é um problema quando os nós folhas já estão cheios
- Nesses casos deve-se reconstruir a árvore, de forma a manter todas as folhas no mesmo nível e com as quantidades corretas de elementos nos demais nós



Inserção



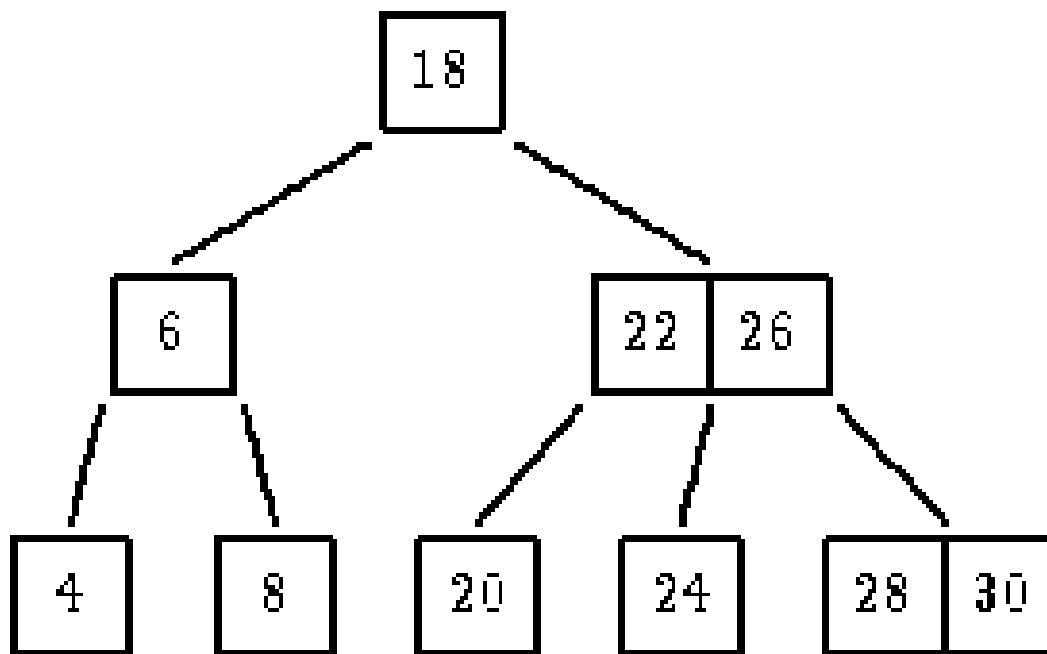
- O processo de reconstrução da árvore envolve divisões de nós, passagem de chaves entre níveis e, possivelmente, a elevação da raiz em um nível



Inserção

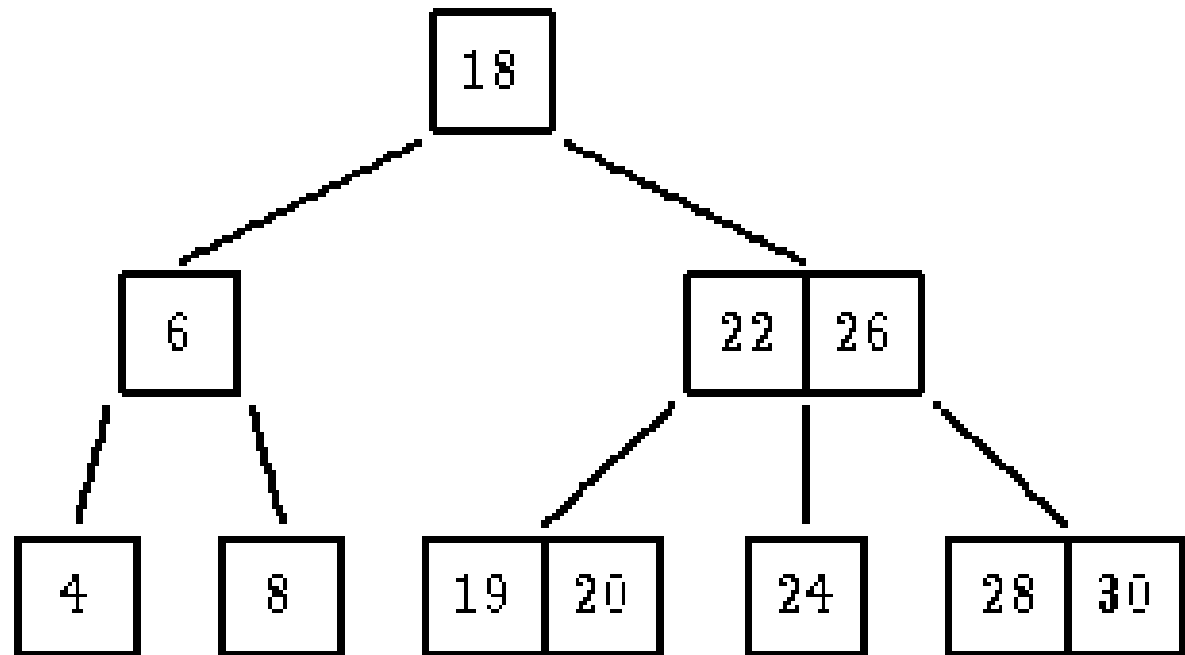


Inserir 19





Inserção

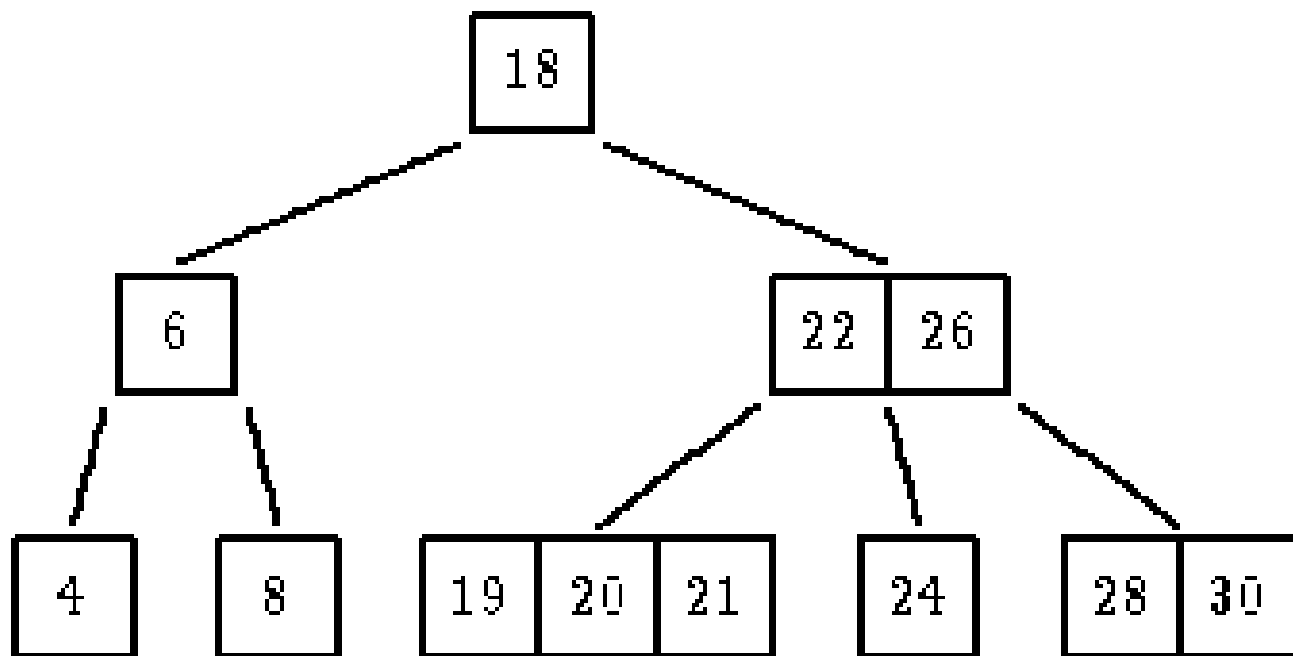




Inserção

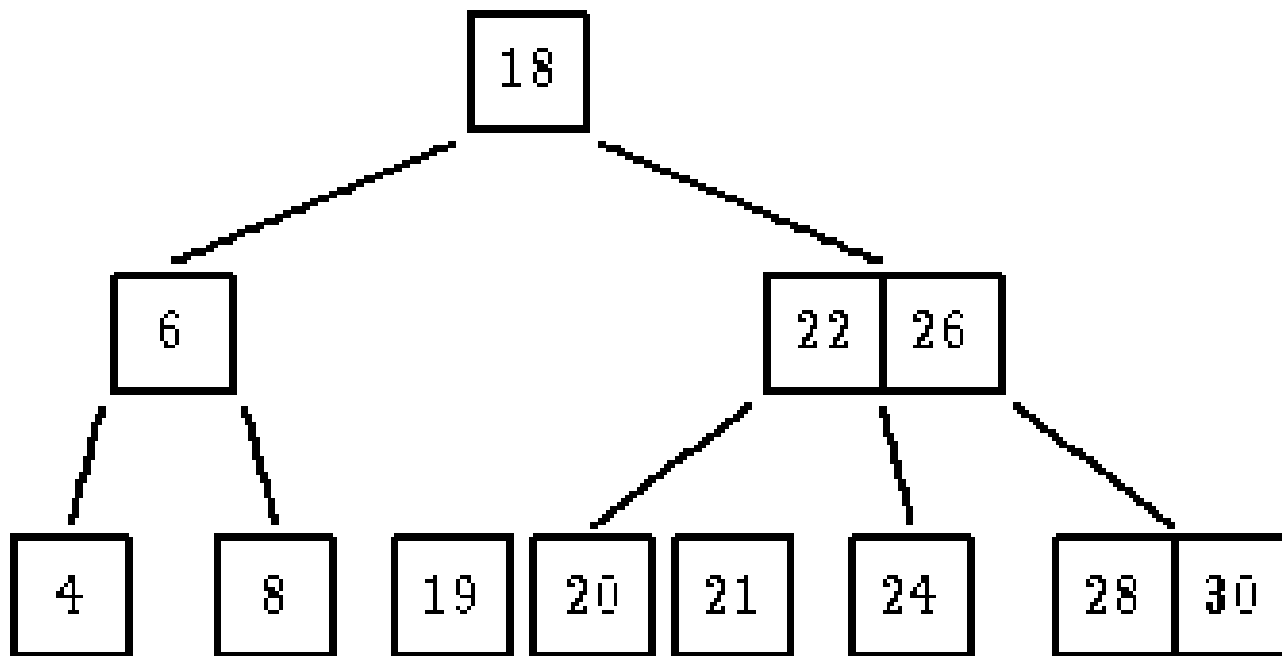


Inserir 21



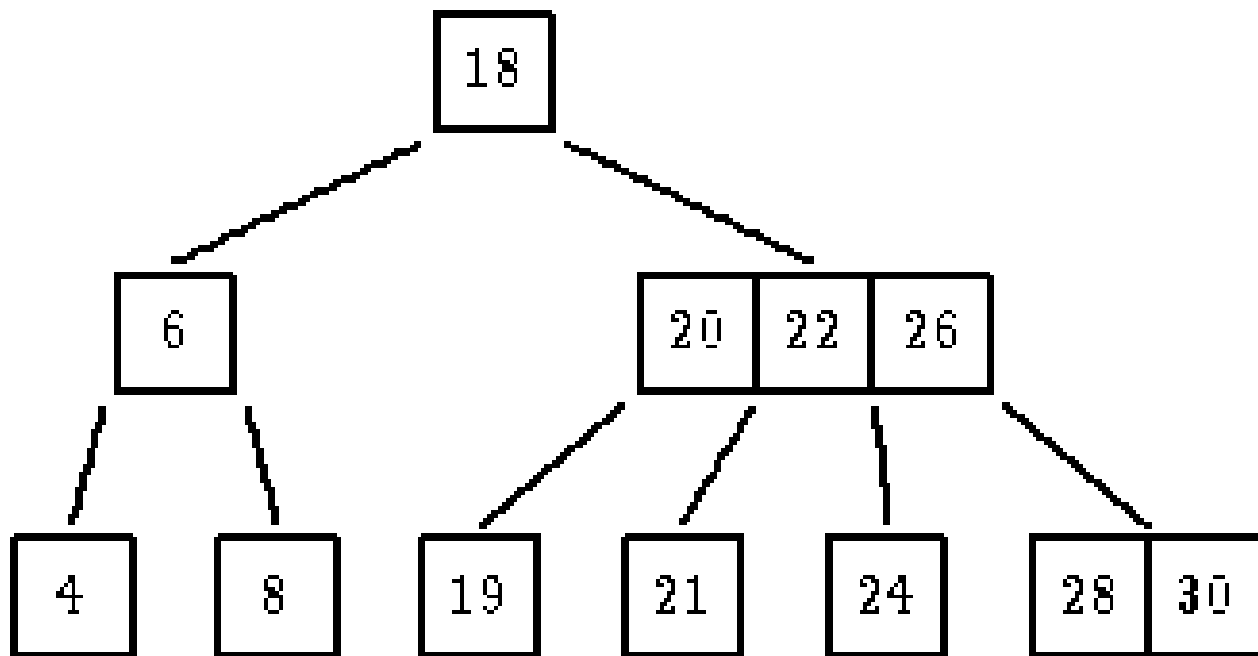


Inserção



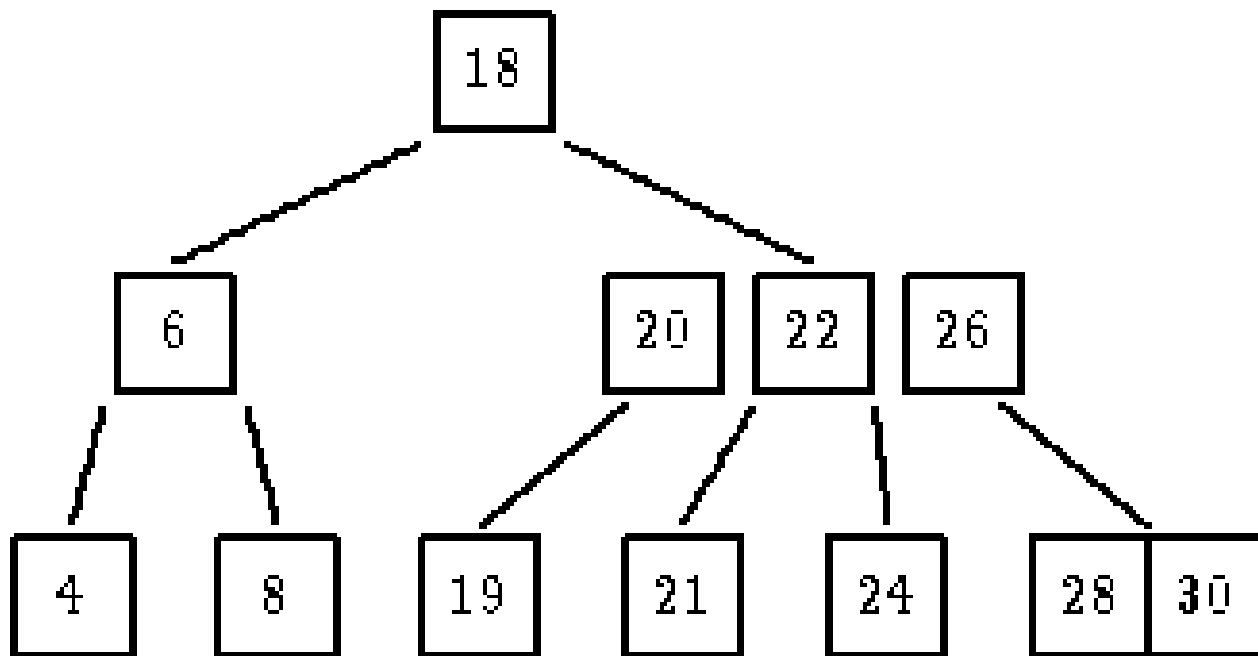


Inserção



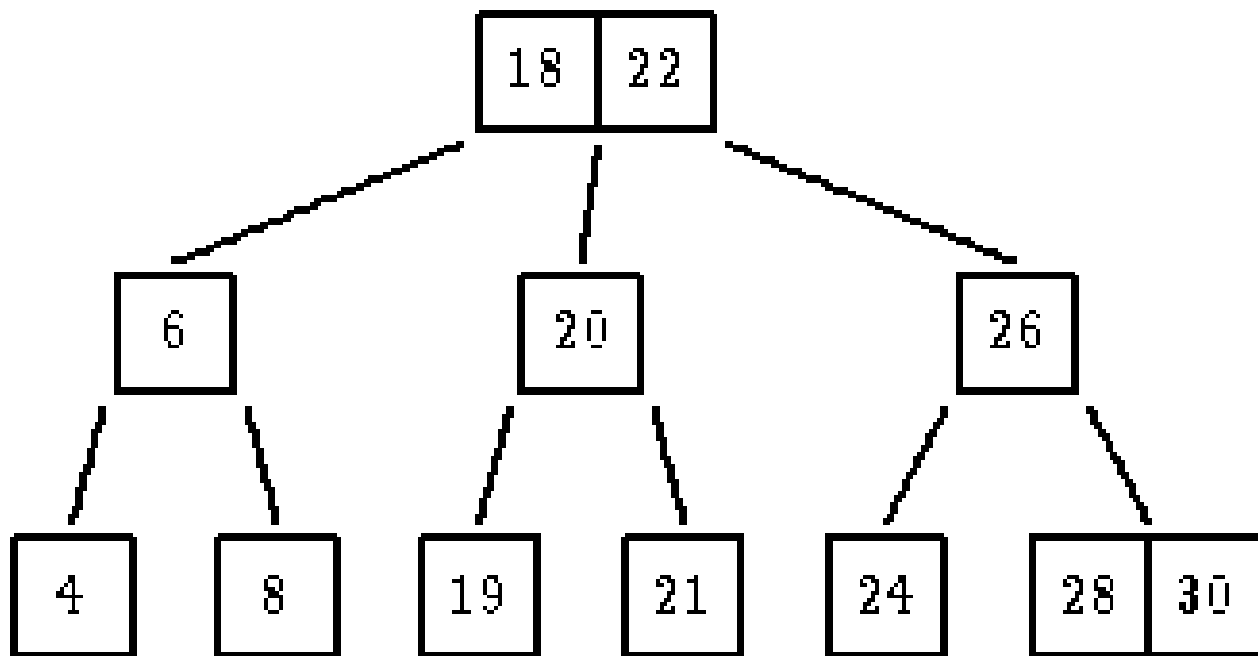


Inserção





Inserção





Remoção em árvores B



- O processo de remoção é equivalente ao da inserção
- Mais uma vez o problema é garantir que após a retirada do elemento as restrições da árvore serão mantidas
- Novamente é preciso movimentar elementos entre nós e níveis da árvore



Remoção em árvores B



- Os casos a serem tratados são:
 - 1) Remoção de elemento de um nó folha
 - 2) Remoção de elemento em nó intermediário ou raiz
 - 3) Remoção de elemento resultando no esvaziamento do nó



Árvores B*



- São variações de árvores B em que se exige que cada nó esteja pelo menos $2/3$ cheio
- A vantagem nessa abordagem é uma melhor ocupação de cada nó e, consequentemente, numa maior quantidade de elementos lidos a cada acesso à memória



Árvores B*



- Essa organização reduz também o volume de movimentações necessárias entre nós e níveis da árvore
- Quando ocorrem, as divisões de nós são feitas com dois nós tornando-se três e não dividindo um em dois, como na árvore B regular



Árvores B+



- Em árvores B ou B* temos problemas na realização de percursos na árvore, uma vez que a passagem de um nó para outro envolve a passagem entre um bloco e outro do disco
- Uma alternativa para esse problema é apresentada pelas árvores B+



Árvores B+



- Nessas árvores os nós intermediários contém apenas indexadores para os dados, contidos nas folhas
- Assim, qualquer busca pode ser feita com poucos acessos ao disco, desde que os demais níveis possam ser armazenados em memória



Árvores 2-3-4-B



- São árvores B de ordem 4
- São também chamadas de árvores 2-4-B
- São usadas como alternativas para as árvores binárias como mecanismos de organização interna de dados



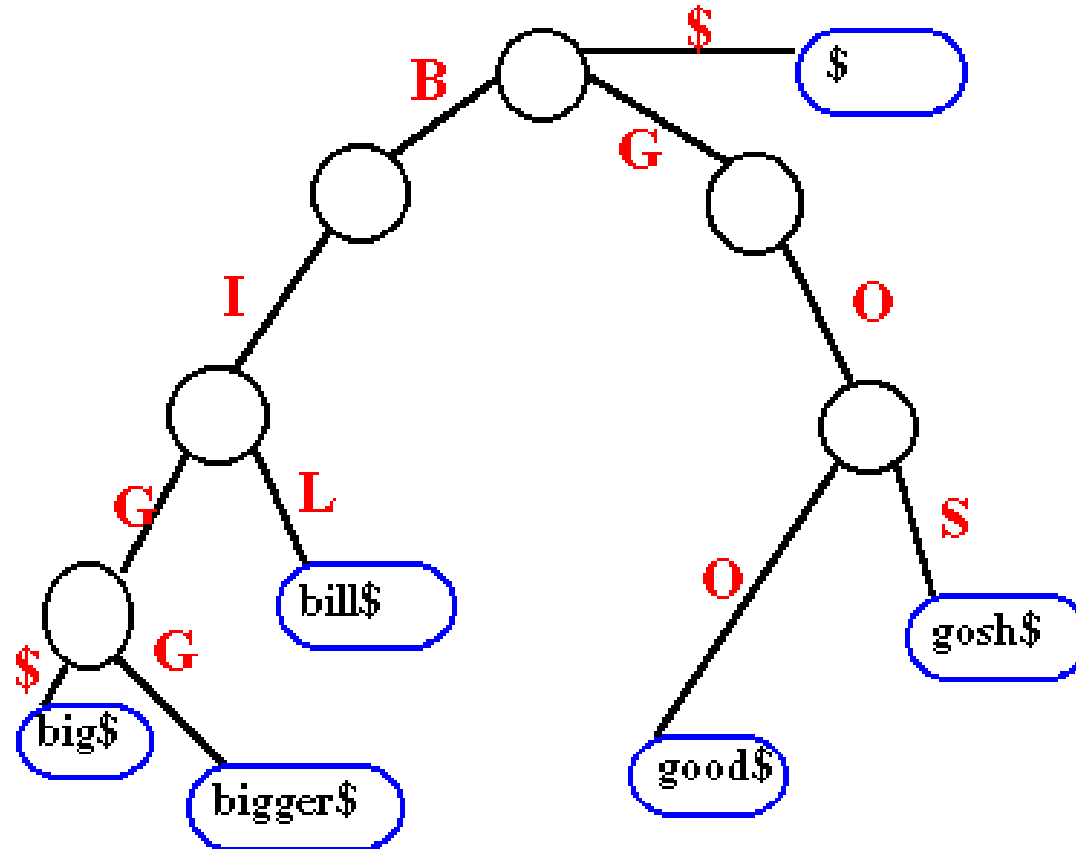
Árvores Trie



- Seu nome vem do termo “retrieval”
- A idéia das árvores trie é usar partes das chaves para guiar o caminho de busca
- Isso as torna muito interessantes na construção de dicionários, como os existentes em tabelas de símbolos dos compiladores



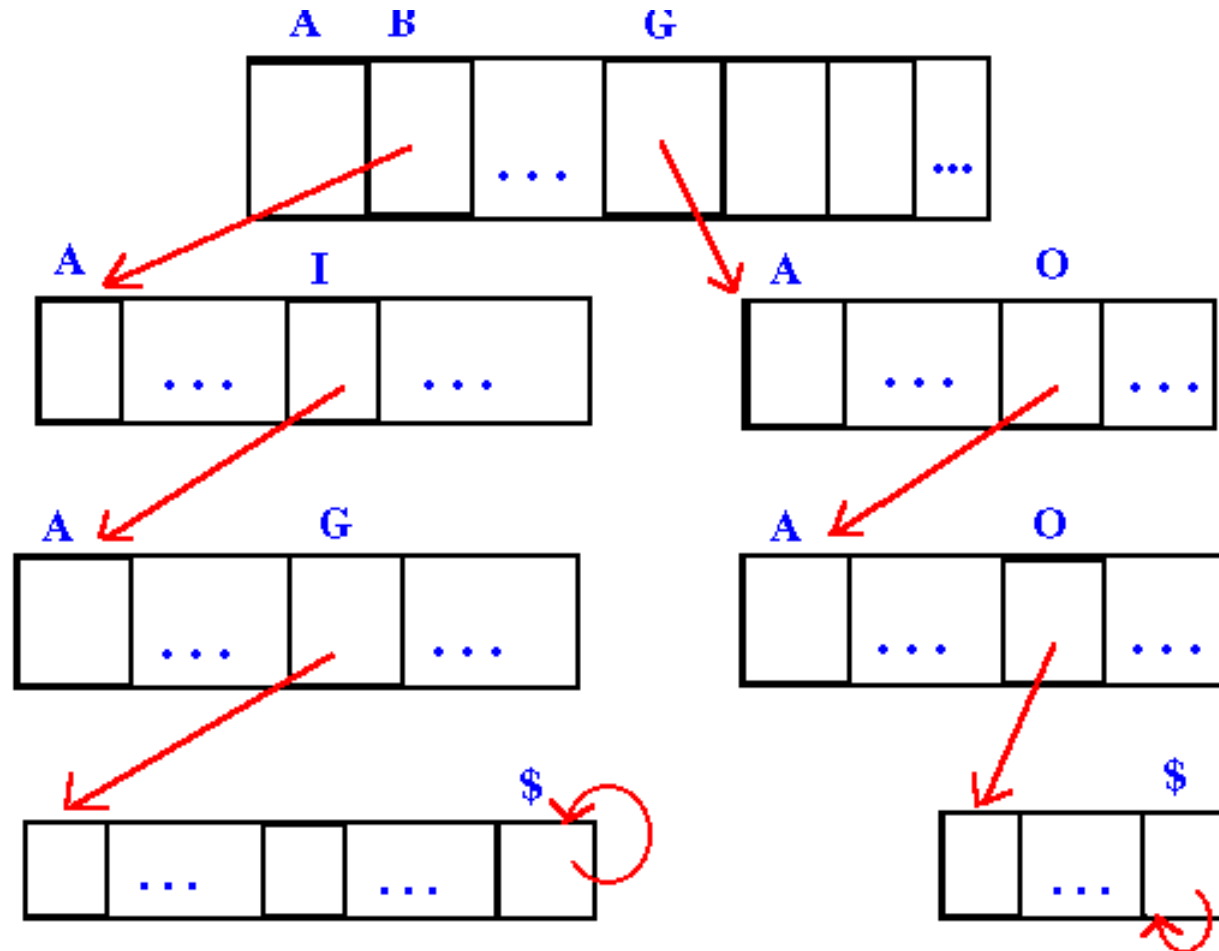
Árvores Trie



Trie for strings big, bigger, bill, good, gosh



Árvores Trie



Trie representation for words **BIG** and **GOO** using array of pointers