b) Aponte a maior sequência de palavras que se repete no banco de dados e mostre como localizar, em qualquer árvore Patricia, esse tipo de ocorrência.

24. Construa, passo a passo, a **árvore Patricia** para as seis primeiras **chaves semi-infinitas** do texto abaixo, representado como uma sequência de *bits*:

0 1 1 0 0 1 1 0 1 1 0 0 1 ··· Texto

1 2 3 4 5 6 7 8 9 ····· Posição

25. Projete e implemente um sistema de programas para recuperação eficiente de informação em bancos de dados constituídos de textos. Tais bancos de dados geralmente recebem adições periódicas, mas nenhuma atualização do que já existe é realizada. Além disso, o tipo de consulta aos dados é totalmente imprevisível. Esses conjuntos de dados aparecem em sistemas legislativos e judiciários, bibliotecas, jornalismo, automação de escritório, entre outros.

Neste trabalho, você deve utilizar um método que cria um índice cuja estrutura é uma **árvore Patricia**, construída a partir de uma sequência de **chaves semi-infinitas**.

O sistema de programas deverá ser capaz de:

a) construir a árvore Patricia sobre um texto de tamanho arbitrário, representado como um conjunto de palavras;

b) ler um conjunto de palavras de tamanho arbitrário;

c) encontrar todas as ocorrências do conjunto de palavras no texto, imprimindo junto com o conjunto algumas palavras anteriores e posteriores no texto;

d) informar o número de ocorrências do conjunto de palavras no texto;

e) encontrar o maior conjunto de palavras que se repete pelo menos uma vez no texto e informar o seu tamanho;

f) dado um inteiro, encontrar, se houver, todas as ocorrências de conjuntos de palavras no texto cujo tamanho seja igual ao inteiro dado.

26. Projete e implemente um sistema de programas para recuperação eficiente de informação em bancos de dados constituídos de textos. Utilize uma estrutura de dados chamada ***Pat array*** (Gonnet e Baeza-Yates, 1991), construída a partir de uma sequência de **chaves semi-infinitas**. O *Pat array* é uma representação compacta da árvore Patricia (vide Seção 5.4), por armazenar apenas os nós externos da árvore. O arranjo é constituído de apontadores para o início de cada palavra de um arquivo de texto. Logo, é necessário apenas um apontador para cada ponto de indexação no texto. Esse arranjo deverá estar indiretamente ordenado pela ordem lexicográfica das chaves semi-infinitas, conforme mostrado na Figura 5.22.

A construção de um *Pat array* é equivalente à ordenação de registros de tamanhos variáveis, representados pelas chaves semi-infinitas. Qualquer operação sobre a árvore Patricia poderá ser simulada sobre o *Pat array* a um custo adicional de $O(\log n)$. Mais ainda, para a operação de pesquisa de prefixo, a árvore Patricia não precisa de fato ser simulada, sendo possível obter algoritmos de custo $O(\log n)$ em vez de $O(\log^2 n)$ para essa operação, a qual pode ser implementada com o emprego de uma pesquisa binária indireta sobre o arranjo, com o resultado de cada comparação sendo menor que, igual ou menor que. *Pat arrays* são também chamados **arranjos de sufixos** (Manber e Myers, 1990).

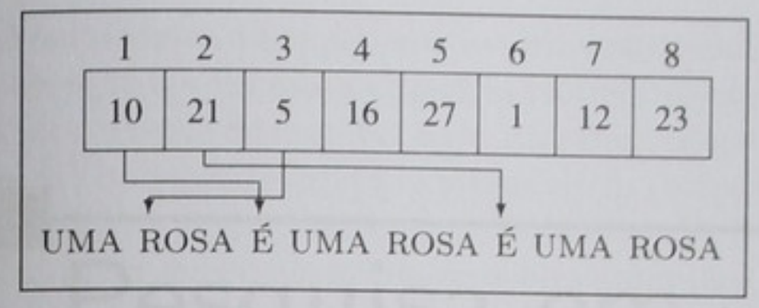

Figura 5.22 Pat array.

O sistema de programas deverá ser capaz de:

a) construir o *PAT array* sobre um texto de tamanho arbitrário, representado como um conjunto de palavras;

b) ler um conjunto de caracteres de tamanho arbitrário. Esse conjunto poderá ser uma palavra ou um prefixo de palavra;

c) informar o número de ocorrências do conjunto de caracteres no texto;

d) encontrar todas as ocorrências do conjunto de caracteres no texto, imprimindo junto com o conjunto algumas palavras anteriores e posteriores no texto;

e) encontrar o maior conjunto de palavras que se repete pelo menos uma vez no texto e informar o seu tamanho;

f) dado um inteiro, encontrar, se houver, todas as ocorrências de conjuntos de palavras no texto cujo tamanho seja igual ao inteiro dado.

A partir disso:

a) apresente a complexidade de pior caso para a letra (c);

b) mostre a relação entre o *PAT array* e a árvore Patricia.

27. Escreva um procedimento recursivo, com base em um dos caminhamentos, para calcular a altura de uma **árvore binária** (Loureiro, 2003).