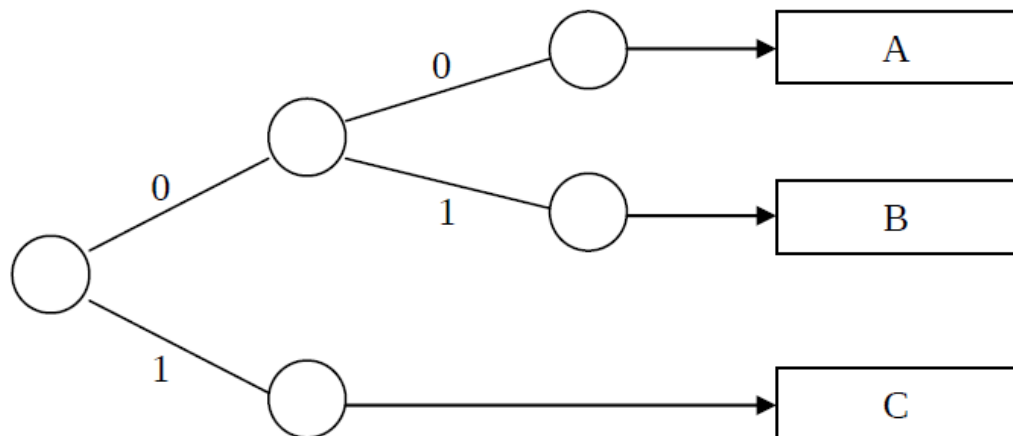


SCC0503 – Algoritmos e Estruturas de Dados II – Lista sobre Hash – Yah!

1. Explique como funciona o *hashing* extensível.
2. Qual a vantagem de usar a representação em diretório no *hashing* extensível, ao invés de usar a representação por árvore da *trie*?
3. As *tries* utilizadas como base do *hashing* extensível têm ordem 2. Como a performance é alterada se aumentarmos essa ordem?
4. Considere um diretório de *hashing* extensível com no máximo 3 elementos por *bucket*, e que a função *hash* gera 4 bits para uma chave. Simule a inserção de chaves que geram os seguintes endereços: 0000, 1000, 1001, 1010, 1100, 0001, 0100, 1111, 1011.
5. Considere a seguinte sequência de chaves: MALUF, QUÉRCIA, SERRA, CARDOSO, LULA, ALVES, GOMES, JEREISSATI, FREIRE, MAGALHÃES, FERREIRA, ANDRADE, CAMARGO. Construa a *trie* adequada para armazenar esse conjunto de chaves.
6. Considere a seguinte *trie* de ordem 2, com ponteiros para *buckets* com capacidade para abrigar 100 chaves (ou registros):



- a) Desenhe a *trie* estendida (árvore binária completa) e o diretório de endereços *hash* correspondente.
- b) Considerando que os *buckets* A, B e C contêm, respectivamente, 100, 50 e 3 registros, dê a configuração do diretório, e a condição de cada *bucket* após a inserção de uma nova chave cujo valor da função *hash* seja 00.
- c) Ainda na configuração inicial, considere agora que todas as chaves de B são eliminadas. O que acontece com o diretório?