**1. Definição e Propriedades de Árvores B+:**

-**Ordem da árvore:** A ordem de uma árvore B+ é definida pelo número máximo de filhos que cada nó pode ter, geralmente indicado como "***m***". Em uma árvore de ordem ***m***, cada nó interno (não-folha) pode ter até ***m-1*** chaves e ***m*** filhos. Uma árvore B+ de ordem ***m = 4*** será utilizada como exemplo. Isso significa que cada nó não-folha pode ter até 3 chaves e 4 filhos.

```

**[30, 60] <- Nó raiz (não-folha)**

**/ | \**

**[10,20] [35,45,55] [70,80,90] <- Nós folha, conectados em sequência**

```

Neste exemplo, as chaves em cada nó não-folha ajudam a localizar rapidamente os intervalos nas folhas.

-**Organização das chaves nos nós folha e não-folha**:

- *Nós folha*: Contêm todas as chaves da árvore e são organizados em ordem crescente. Cada chave está associada a um ponteiro para um dado ou registro.

- *Nós não-folha*: Servem apenas para roteamento, contendo chaves que indicam intervalos. Essas chaves ajudam a direcionar a busca até os nós folha.

- **Ligação sequencial entre as folhas**: As folhas são conectadas sequencialmente, permitindo acesso rápido aos dados de forma ordenada, o que é ideal para operações de intervalo.

- **Balanceamento da árvore**: A árvore B+ é balanceada, pois todas as folhas estão no mesmo nível. O balanceamento é mantido durante inserções e remoções por meio de operações de cisão (split) e redistribuição de nós quando um nó excede sua capacidade ou fica abaixo do limite mínimo de chaves.

**2. Inserção em uma Árvore B+:**

Ao inserir as chaves em uma árvore B+ de ordem 4. Em cada etapa, verificamos se o nó ultrapassa o limite máximo de chaves, realizando cisões (*splits*) quando necessário. Aqui está o processo de inserção das chaves na ordem fornecida (45, 12, 33, 50, 27, 18, 60, 72, 15), com um resumo de cada etapa:

1. *Inserir 45*: Como a árvore está vazia, 45 é inserido na folha.

```

**[45]**

```

2. *Inserir 12*: 12 é adicionado na mesma folha, com as chaves ficando ordenadas (12, 45).

```

**[12, 45]**

```

3. *Inserir 33*: A folha acomoda 33, mantendo a ordem (12, 33, 45).

```

**[12, 33, 45]**

```

4. *Inserir 50*: Com o limite da folha atingido, ocorre uma cisão. A folha se divide, com a chave 33 promovida ao nível superior.

```

**[33]**

**/ \**

**[12] [45, 50]**

```

5. *Inserir 27*: 27 é inserido no nó folha adequado (12, 27).

```

**[33]**

**/ \**

**[12, 27] [45, 50]**

```

6. *Inserir 18*: Inserido no mesmo nó folha, ficando (12, 18, 27).

```

**[33]**

**/ \**

**[12, 18, 27] [45, 50]**

```

7. *Inserir 60*: Inserido na folha correta (45, 50, 60).

```

**[33]**

**/ \**

**[12, 18, 27] [45, 50, 60]**

```

8. *Inserir 72*: Causa uma cisão no nó (45, 50, 60, 72), com 60 promovido.

```

**[33, 60]**

**/ | \**

**[12, 18, 27] [45, 50] [72]**

```

9. *Inserir 15*: Inserido no nó folha adequado, resultando em redistribuição ou nova cisão se necessário.

```

**[33, 60]**

**/ | \**

**[12, 15, 18, 27] [45, 50] [72]**

```

Após cada inserção, o balanceamento é mantido através das operações de cisão, e as folhas são redistribuídas conforme o limite de ordem.

**3. Remoção em uma Árvore B+:**

Para a árvore resultante da etapa anterior, seguimos as remoções das chaves na ordem indicada (50, 18, 12). Ou seja, usarei a árvore resultante da inserção e remover as chaves 50, 18 e 12:

1. *Remover 50*: É removido diretamente da folha; se houver redistribuição, ela ocorre entre folhas vizinhas.

```

**[33, 60]**

**/ | \**

**[12, 15, 18, 27] [45] [72]**

```

2. *Remover 18*: Removido da folha; redistribuição ou concatenação ocorre se o nó ficar abaixo do limite mínimo.

```

**[33, 60]**

**/ | \**

**[12, 15, 27] [45] [72]**

```

3. *Remover 12*: Remoção segue a mesma lógica, mantendo o balanceamento.

```

**[33, 60]**

**/ | \**

**[15, 27] [45] [72]**

```

Após cada remoção, se um nó fica com poucas chaves, ocorre redistribuição ou concatenação com nós irmãos, mantendo as propriedades da árvore B+.

**4. Busca em uma Árvore B+:**

- *Buscar a chave 72*:

- Começamos no nó raiz `**[33, 60]**`.

- Como 72 é maior que 60, seguimos pelo terceiro ponteiro, chegando ao nó `**[72]**` e encontramos a chave.

- *Buscar a chave 25*:

- Começamos no nó raiz `**[33, 60]**`.

- Como 25 é menor que 33, seguimos pelo primeiro ponteiro, chegando ao nó `**[15, 27]**`.

- Como 25 não está presente nesse nó, concluímos que a chave não está na árvore.

### Questão 3: Remoção na Árvore B+

Vamos usar a árvore resultante da inserção e remover as chaves 50, 18 e 12.

#### Remoção passo a passo:

1. \*\*Remover 50\*\*

2. \*\*Remover 18\*\*

3. \*\*Remover 12\*\*

### Questão 4: Busca na Árvore B+

Para realizar as buscas de 72 e 25, percorremos a árvore:

- \*\*Buscar 72\*\*:

- \*\*Buscar 25\*\*:

Essas ilustrações descrevem como as operações de inserção, remoção e busca são realizadas na estrutura da árvore B+.