

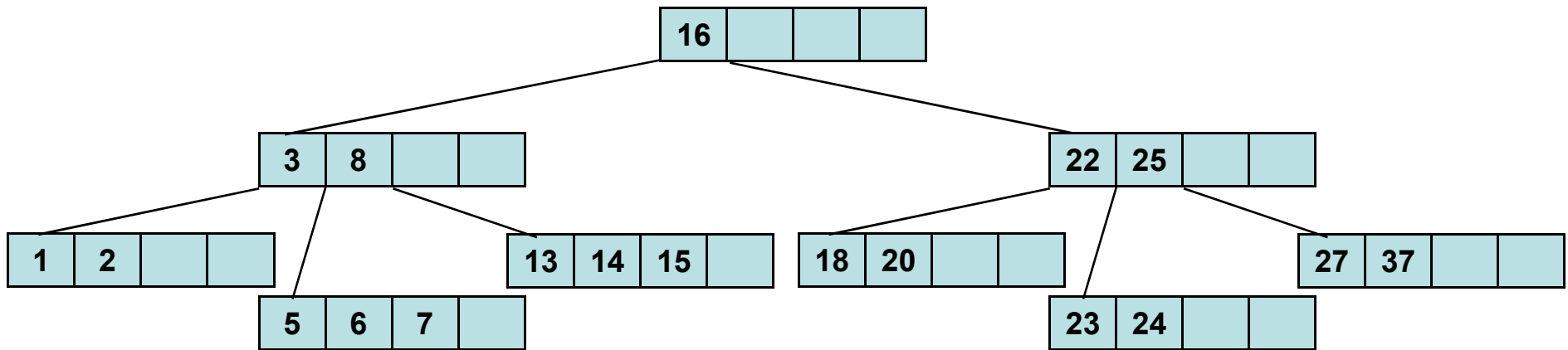
Árvores B

Veremos agora o processo de remoção de uma chave em uma árvore B. Grande parte deste processo pode ser vista como o inverso da inserção, claro que com suas particularidades.

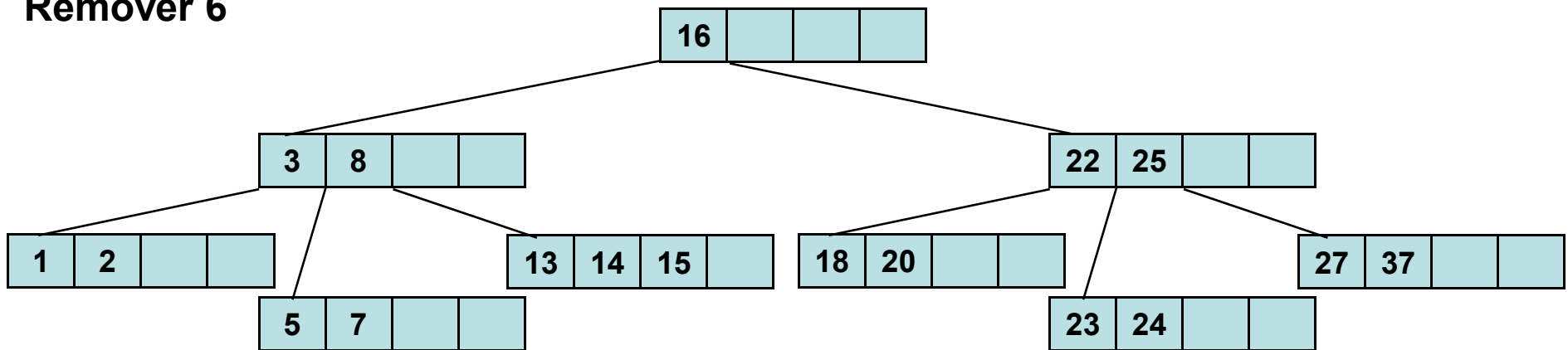
Basicamente, podemos dividir o processo de remoção em dois casos principais: remover uma chave de uma folha e remover uma chave de um nó não-folha. Veremos agora o detalhamento do processo.

1. Remover uma chave de uma folha.

1.1. Se, depois de remover uma chave K a folha está pelo menos metade cheia e somente chaves maiores do que K são movidas para a esquerda do buraco (como na árvore do slide a seguir). É o inverso do caso 1 de inserção.

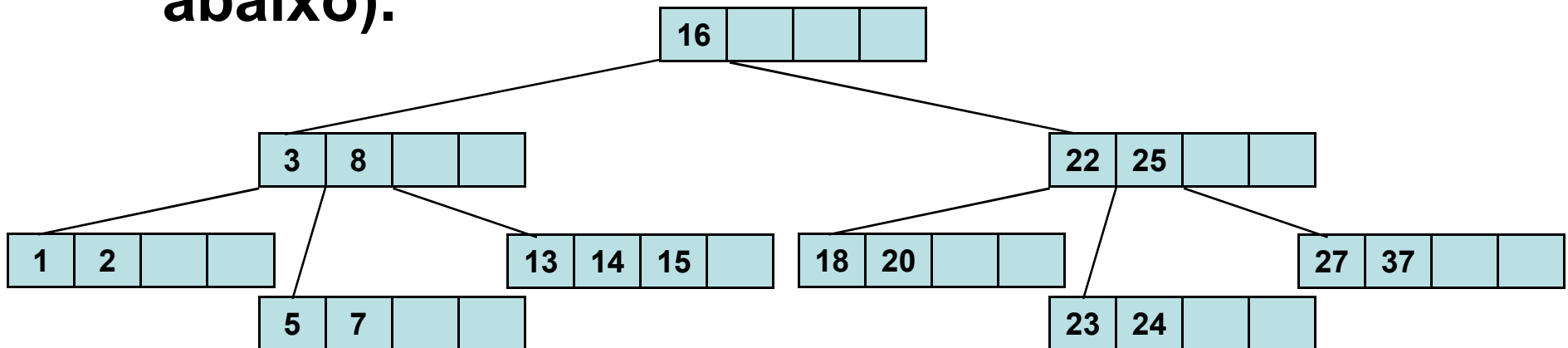


Remover 6

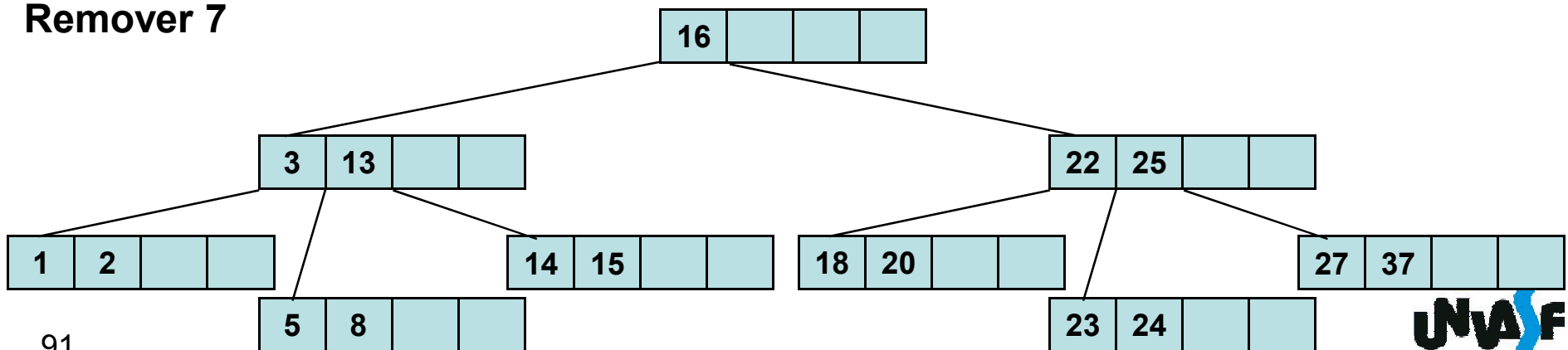


1.2. Se, depois de remover K, o número de chaves na folha é menor que $\lceil m/2 \rceil - 1$, causando uma subutilização.

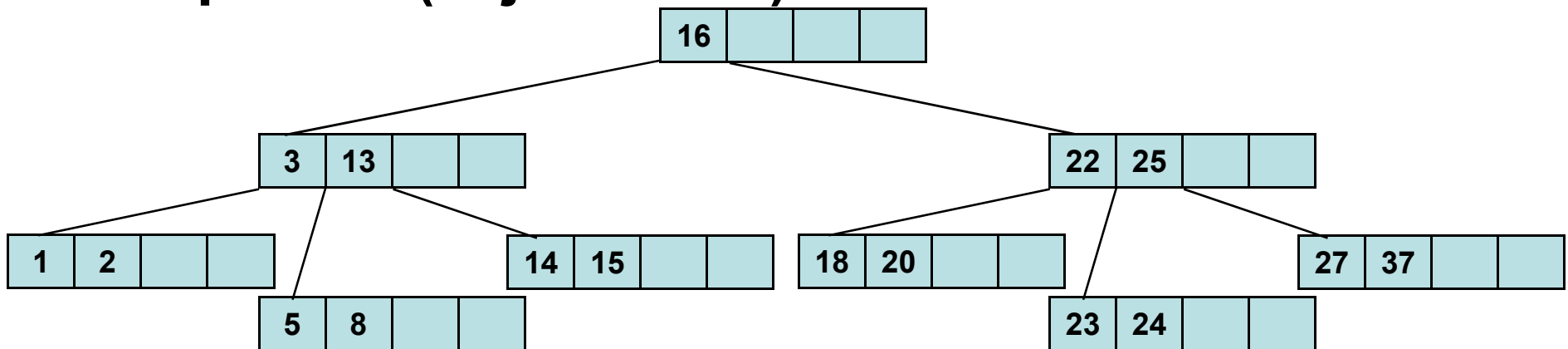
1.2.1. Se existe um irmão à esquerda ou à direita com o número de chaves excedendo o mínimo $\lceil m/2 \rceil - 1$, então todas as chaves desta folha e deste irmão são redistribuídas entre eles, movendo-se a chave separadora do ascendente até a folha e movendo-se uma chave do irmão até o ascendente (veja abaixo).



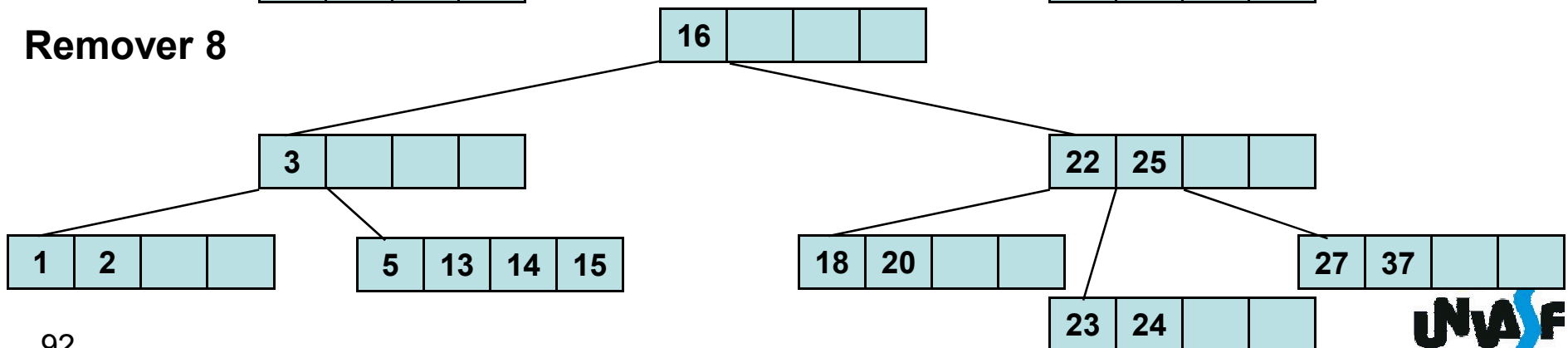
Remover 7



1.2.2. Se a folha é subutilizada e o número de chaves em seu(s) irmão(s) é $\lceil m/2 \rceil - 1$, a folha e um irmão são fundidos, as chaves da folha, de seu irmão e a chave de separação do ascendente são todas colocadas na folha e o nó irmão é descartado. As chaves no ascendentes são movidas se um buraco aparece (veja abaixo).

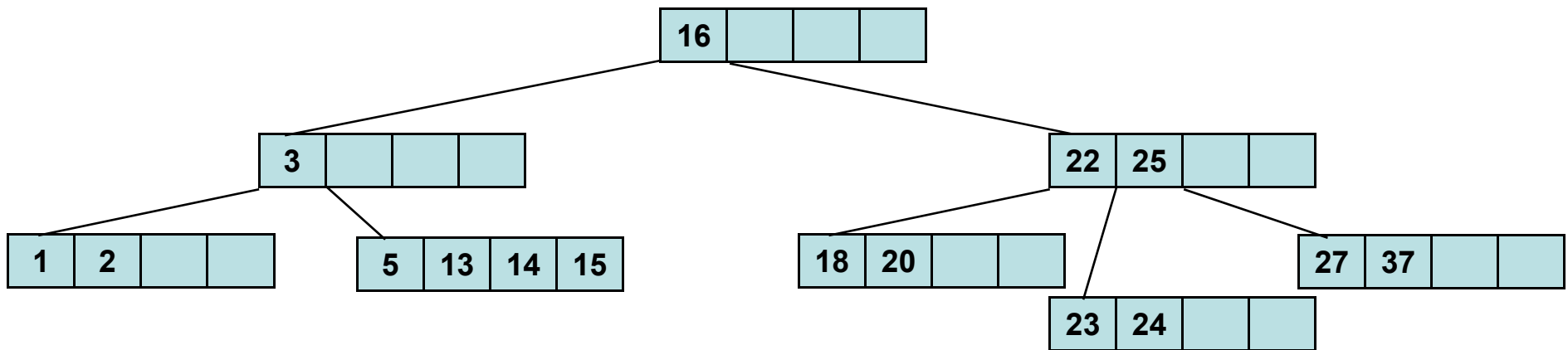


Remover 8

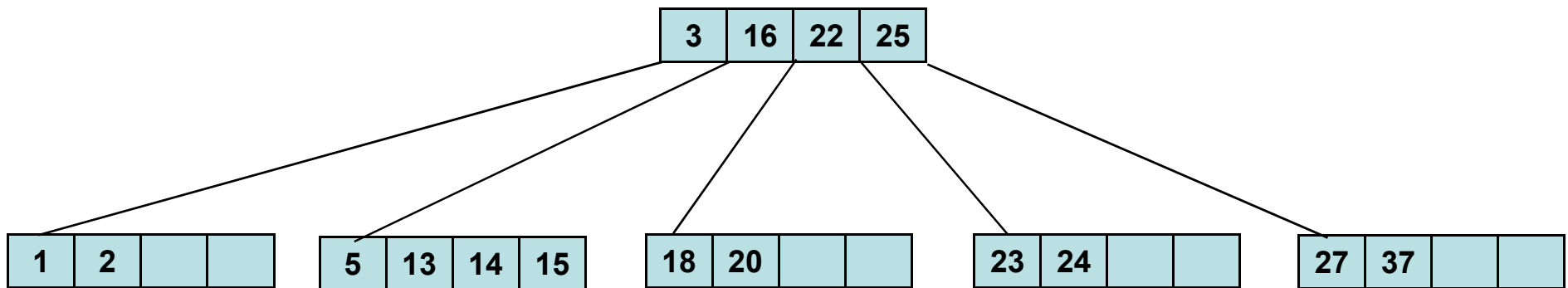


Isso pode inicializar uma cadeia de operações se o ascendente é subutilizado. O ascendente é agora tratado como se ele fosse uma folha, e a etapa 1.2.2 é repetida até que a etapa 1.2.1 possa ser executada ou a raiz da árvore tenha sido atingida. Isso é o inverso do caso 2 de inserção.

1.2.2.1. Um caso particular resulta em fundir uma folha ou não folha com seu irmão quando seu ascendente é a raiz com somente uma chave. Neste caso, as chaves do nó e seu irmão, junto com a única chave da raiz, são colocadas no nó que se torna uma nova raiz, e tanto o nó irmão como o nó raiz velho são descartados. É o único caso em que dois nós desaparecem ao mesmo tempo. Além disso, a altura da árvore é diminuída por um (veja este processo no próximo slide). Isso é o inverso do caso 3 de inserção.

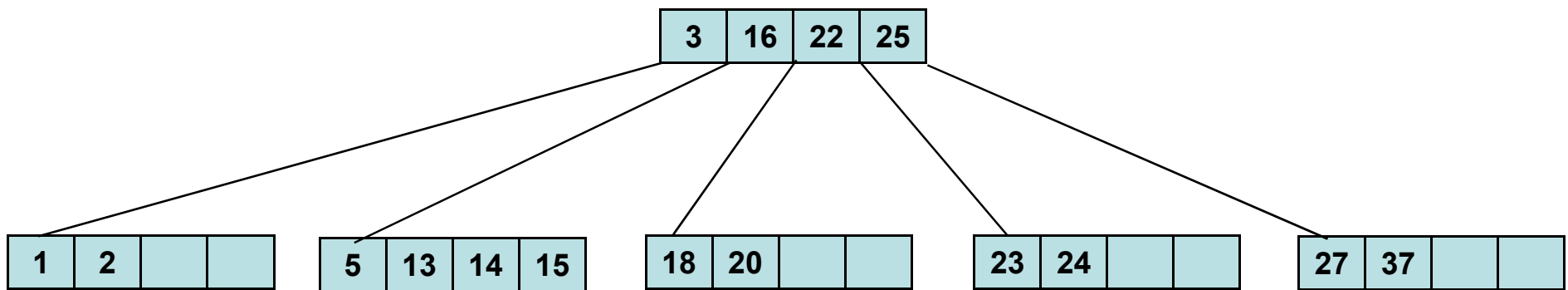


Remover 8 (continuação)

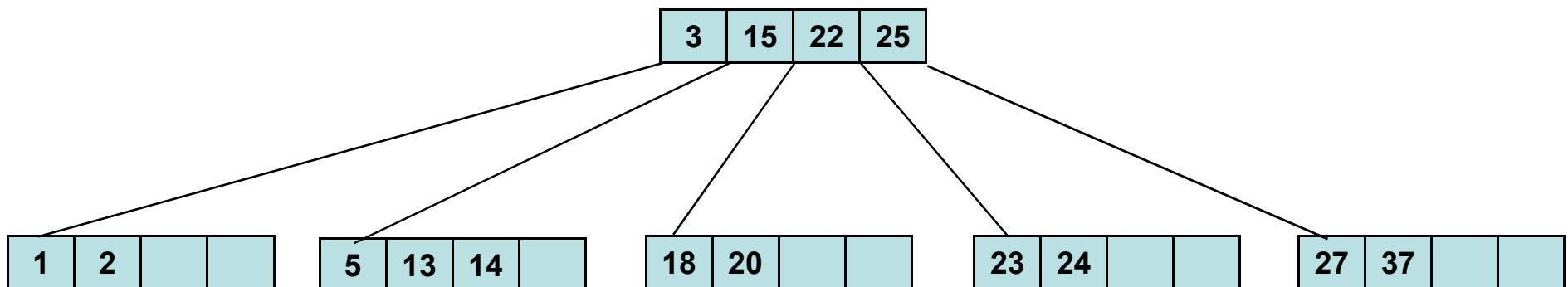


2. Remover uma chave de uma não folha. Isso pode levar a problemas com a organização da árvore. Em consequência, a remoção de um nó não folha é transformada na remoção de uma chave de uma folha.

A chave a ser removida é substituída por seu antecessor imediato (o sucessor poderia também ser usado), que pode ser encontrado somente em uma folha. Essa Chave antecessora é removida da folha, o que nos trás para o caso 1 anterior (veja abaixo).



Remover 16



Árvores B

Com base no procedimento descrito elabore um algoritmo em alto nível, representando-o através de um pseudocódigo, que descreva o processo de remoção.

```
vazio remocaoEmArvoreB (TipoChave K, ArvoreB root)
nó = buscaEmArvoreB (K, root);
se (nó != NULL) então
    se (nó não é uma folha) então
        Encontre uma folha com antecessor S mais
        próximo de K;
        Copia S sobre K no nó;
        nó = a folha que contém S;
        Remova S de nó;
    senão
        Remova K de nó;
fimse
enquanto (1)
    se (nó não esta subutilizado) então
        retorne;
```



```

senão
então se (há um irmão de nó com chaves suficientes)
    Redistribua as chaves entre nó e seu irmão;
    retorne;
senão
    se (o ascendente de nó é a raiz) então
    se (o ascendente tem somente uma chave)
então
    Faça a fusão de nó, seu irmão e o
    ascendente para formar uma nova raiz;
senão
    Faça a fusão de nó e seu irmão;
fimse
retorne;
senão
    Faça a fusão de nó e seu irmão;
    nó = seu ascendente;
fimse
fimse
fimse
fimenquanto
fimse
fim

```

Árvores B

Com base no algoritmo apresentado elabore uma função em C que execute a operação em questão.