

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_KEYS 3 // Máximo de chaves por nó (pode ser ajustado)
#define MIN_KEYS (MAX_KEYS / 2)

typedef struct {
    char nome[50];
    char estado[3];
    char regioao[20];
} Aeroporto;

typedef struct BTreeNode {
    Aeroporto chaves[MAX_KEYS];
    struct BTreeNode* filhos[MAX_KEYS + 1];
    int numChaves;
    int folha;
} BTreeNode;

typedef struct {
    BTreeNode* raiz;
} BTree;

BTreeNode* criarNo(int folha) {
    BTreeNode* novoNo = (BTreeNode*)malloc(sizeof(BTreeNode));
    novoNo->folha = folha;
    novoNo->numChaves = 0;
    for (int i = 0; i < MAX_KEYS + 1; i++) {
        novoNo->filhos[i] = NULL;
    }
    return novoNo;
}

BTree* criarArvoreB() {
    BTree* arvore = (BTree*)malloc(sizeof(BTree));
    arvore->raiz = criarNo(1); // A raiz começa como folha
    return arvore;
}

void splitFilho(BTreeNode* no, int i, BTreeNode* filho) {
    BTreeNode* novoFilho = criarNo(filho->folha);
    novoFilho->numChaves = MIN_KEYS;

    for (int j = 0; j < MIN_KEYS; j++) {
        novoFilho->chaves[j] = filho->chaves[j + MIN_KEYS + 1];
    }

    if (!filho->folha) {
        for (int j = 0; j < MIN_KEYS + 1; j++) {
            novoFilho->filhos[j] = filho->filhos[j + MIN_KEYS + 1];
        }
    }
}
```

```
}

filho->numChaves = MIN_KEYS;

for (int j = no->numChaves; j >= i + 1; j--) {
    no->filhos[j + 1] = no->filhos[j];
}

no->filhos[i + 1] = novoFilho;

for (int j = no->numChaves - 1; j >= i; j--) {
    no->chaves[j + 1] = no->chaves[j];
}

no->chaves[i] = filho->chaves[MIN_KEYS];
no->numChaves++;
}

void inserirNaoCheio(BTreeNode* no, Aeroporto aeroporto) {
    int i = no->numChaves - 1;

    if (no->folha) {
        while (i >= 0 && strcmp(aeroporto.nome, no->chaves[i].nome) < 0) {
            no->chaves[i + 1] = no->chaves[i];
            i--;
        }
        no->chaves[i + 1] = aeroporto;
        no->numChaves++;
    } else {
        while (i >= 0 && strcmp(aeroporto.nome, no->chaves[i].nome) < 0) {
            i--;
        }
        i++;
        if (no->filhos[i]->numChaves == MAX_KEYS) {
            splitFilho(no, i, no->filhos[i]);
            if (strcmp(aeroporto.nome, no->chaves[i].nome) > 0) {
                i++;
            }
        }
        inserirNaoCheio(no->filhos[i], aeroporto);
    }
}

void inserirAeroporto(BTree* arvore, Aeroporto aeroporto) {
    BTreeNode* raiz = arvore->raiz;
    if (raiz->numChaves == MAX_KEYS) {
        BTreeNode* novoNo = criarNo(0);
        novoNo->filhos[0] = raiz;
        splitFilho(novoNo, 0, raiz);
        arvore->raiz = novoNo;
        inserirNaoCheio(novoNo, aeroporto);
    } else {
        inserirNaoCheio(raiz, aeroporto);
    }
}
```

```
}

void buscarAeroporto(BTreeNode* no, char* nome) {
    int i = 0;
    while (i < no->numChaves && strcmp(nome, no->chaves[i].nome) > 0) {
        i++;
    }
    if (i < no->numChaves && strcmp(nome, no->chaves[i].nome) == 0) {
        printf("Aeroporto: %s\nEstado: %s\nRegiao: %s\n", no->chaves[i].nome, no->chaves[i].estado, no->chaves[i].regiao);
        return;
    }
    if (no->folha) {
        printf("Aeroporto não encontrado.\n");
        return;
    }
    buscarAeroporto(no->filhos[i], nome);
}

int main() {
    BTree* arvore = criarArvoreB();

    // Exemplo de inserção de aeroportos
    Aeroporto a1 = {"Aeroporto Internacional de Brasilia", "DF", "Brasilia"};
    Aeroporto a2 = {"Aeroporto Internacional do Rio de Janeiro", "RJ", "Rio de Janeiro"};
    Aeroporto a3 = {"Aeroporto Internacional de Manaus", "AM", "Amazonica"};
    Aeroporto a4 = {"Aeroporto Internacional de Recife", "PE", "Recife"};
    Aeroporto a5 = {"Aeroporto Internacional de Curitiba", "PR", "Curitiba"};

    inserirAeroporto(arvore, a1);
    inserirAeroporto(arvore, a2);
    inserirAeroporto(arvore, a3);
    inserirAeroporto(arvore, a4);
    inserirAeroporto(arvore, a5);

    // Exemplo de busca
    buscarAeroporto(arvore->raiz, "Aeroporto Internacional de Brasilia");
    buscarAeroporto(arvore->raiz, "Aeroporto Internacional de Manaus");
    buscarAeroporto(arvore->raiz, "Aeroporto Internacional de Curitiba");

    return 0;
}
```