

```

100 }
101
102 ∨ Arvore *rotacaoEsquerda(Arvore *no) {
103     Arvore *noDir = no->dir;
104     Arvore *noDirEsq = noDir->esq;
105
106     noDir->esq = no;
107     no->dir = noDirEsq;
108
109     return noDir;
110 }
111
112 ∨ Arvore *rotacaoDireita(Arvore *no) {
113     Arvore *noEsq = no->esq;
114     Arvore *noEsqDir = noEsq->dir;
115
116     noEsq->dir = no;
117     no->esq = noEsqDir;
118
119     return noEsq;
120 }
121
122 ∨ void balanciarArvore(Arvore **pNo) {
123 ∨     if (*pNo != NULL) {
124         balanciarArvore(&(*pNo)->esq);
125         balanciarArvore(&(*pNo)->dir);
126
127         int alturaEsq = alturaArvore((*pNo)->esq);
128         int alturaDir = alturaArvore((*pNo)->dir);
129
130 ∨         if (alturaEsq - alturaDir > 1) {
131 ∨             if (alturaArvore((*pNo)->esq->esq) >= alturaArvore((*pNo)->esq->dir)) {
132                 *pNo = rotacaoEsquerda(*pNo);
133 ∨             } else {
134                 *pNo = rotacaoDireita(rotacaoEsquerda(*pNo));
135             }
136 ∨         } else if (alturaDir - alturaEsq > 1) {
137 ∨             if (alturaArvore((*pNo)->dir->dir) >= alturaArvore((*pNo)->dir->esq)) {
138                 *pNo = rotacaoDireita(*pNo);
139 ∨             } else {
140                 *pNo = rotacaoEsquerda(rotacaoDireita(*pNo));
141             }
142         }
143     }
144 }
145

```