CS106X

Autumn 2015 Cynthia Lee

Section 6 (Week 7) – SOLUTION

Problem and solution authors include Jerry Cain and Cynthia Lee.

Problem 1 Solution: Tree Rotations

a)

```
static void rotateRightChildLeft(node **toRotateAddr) {
   node *toRotate = *toRotateAddr;
   node *rightChild = toRotate->right;
   toRotate->right = rightChild->left;
   rightChild->left = toRotate;
   *toRotateAddr = rightChild;
}
```

b)

```
static void pullToRoot(node **rootAddr, int value) {
   node *root = * rootAddr;
   if ((root == NULL) ||
       (root->value == value)) return; // no reason to rotate
   if (root->value < value) { // value would be in right subtree
      pullToRoot(&root->right, value);
      rotateLeft(rootp);
   } else {
      pullToRoot(&root->left, value);
      rotateRight(rootp);
   }
}
```

Problem 2 Solution: Quadtrees

```
static quadtree *gridToQuadtree(Grid<bool>& image,
                                int lowx, int highx, int lowy, int highy) {
   quadtree *qt = new quadtree;
   qt->lowx = lowx; qt->highx = highx - 1;
   qt->lowy = lowy; qt->highy = highy - 1;
   if (allPixelsAreTheSameColor(image, lowx, highx, lowy, highy)) {
      qt->isBlack = image[lowx][lowy];
      for (int i = 0; i < 4; i++) {
         qt->children[i] = NULL;
      }
   } else {
      int midx = (highx - lowx) / 2;
      int midy = (highy - lowy) / 2;
      qt->children[NW] = gridToQuadtree(image, lowx, midx, midy, highy);
      qt->children[NE] = gridToQuadtree(image, midx, highx, midy, highy);
      qt->children[SE] = gridToQuadtree(image, midx, highx, lowy, midy);
      qt->children[SW] = gridToQuadtree(image, lowx, midx, lowy, midy);
```

```
            // assume NW, NE, etc are constants/#defines
        }

        return qt;
    }

    static quadtree *gridToQuadtree(Grid<bool>& image) {
        return gridToQuadtree(image, 0, image.numCols(), 0, image.numRows());
    }
```

(where `allPixelsAreTheSameColor` does exactly what it sounds like it does)