设 B 表示网格图边界上的结点集合，即最外层行和列。假设 G 具有性质 $(*)$，如果边界上与 B 相邻的所有结点 $v \notin B$ 都满足 $v \prec B$。请注意，在一个满足性质 $(*)$ 的网格图 G 中，全局最小值不位于边界上（因为全局最小值小于所有邻居，但位于边界上的结点总有一个邻居在 B 中）——因此 G 至少有一个不位于边界上的局部最小值。我们将这种局部最小值称为内部局部最小值。

现在我们描述一个递归算法，它接受一个满足性质 $(*)$ 的网格图并返回一个内部局部最小值，使用 $O(n)$ 次探索。在算法结束时，我们将描述如何将其轻松转换为总体问题的解决方案。

假设 G 满足性质 $(*)$，并且设 $v \notin B$ 是 B 中一个邻居的结点。设 C 表示中间行和中间列中的结点集合。设 $S = B \cup C$；从 G 中删除 S 将 G 分为四个子网格。最后，设 T 是所有结点都属于 S 的结点集合。

使用 $O(n)$ 次探索，我们在 $S \cup T$ 中找到标记值最小的结点 u。我们知道 $u \notin B$ 且 $v \prec B$。我们有两种情况。如果 $u \in C$，那么 u 是一个内部局部最小值，因为它的所有邻居都在 $S \cup T$ 中，并且都比它大。否则，$u \in T$。设 G' 为包含 u 的子网格，以及它所边界的 S 的部分。现在，G' 满足性质 $(*)$，因为 u 与 G' 的边界结点相邻，并且比 G' 边界上的所有结点都小。因此，G' 有一个内部局部最小值，它也是 G 的一个内部局部最小值。我们递归地在 G' 上调用我们的算法来找到一个内部局部最小值。

如果 $T(n)$ 表示算法找到一个 $n \times n$ 网格中的内部局部最小值所需的探索次数，我们有递推关系 $T(n) = O(n) + T(n/2)$，其解为 $T(n) = O(n)$。

最后，我们将其转换为找到网格 G 的局部最小值（不一定是内部的）的算法。使用 $O(n)$ 次探索，我们在边界 B 上找到标记值最小的结点 v。如果 v 是一个角结点，它就是一个局部最小值，我们完成了。否则，v 有一个唯一不在 B 上的邻居 u。如果 $v < u$，那么 v 是一个局部最小值，我们完成了。否则，G 满足性质 $(*)$（因为 v 是边界上的最小值，且 $v \geq u$）。我们调用上述算法。