Theorem 3.4: Algorithm $PATH1$ solves the max-min k-partitioning problem on a path of n vertices in $O(n)$ time.

Proof of Correctness: Arrays *next* and *ncut*, indexed by vertices on the path, is the key difference between $PATH0$ and $PATH1$. Feasibility test $FTEST1$ requires $next(a)$ to point to a further vertex, b, in the subpath, such that $ncut(a)$ is the number of cuts between a and b. Enforcing this requirement, arrays *next* and *ncut* are updated by two subroutines of $PATH1$.

The first is *glue_paths*. When $small(M) \geq \lambda_2$ or $large(M) \leq \lambda_1$, certain vertices on the subpaths need no longer be inspected, so *glue_paths* combines two adjacent subpaths of equal length by updating the *next* and *ncut* values on the subpaths. The gluing and updating is repeated until no longer possible.

The second subroutine is *compress_next_path*, which is called after *search_next_path*(a, b) in $FTEST1$. Procedure *search_next_path*(a, b) determines $(v, sumcut)$, where v is the location of the final cut before b and *sumcut* is the number of cuts between a and v. The values of *next* and *ncut* for vertices between a and v are then updated.

The correctness of $FTEST1$ follows, as *numcut* is incremented once for each subpath whose weight exceeds λ, or by *sumcut* for each compressed subpath with the corresponding number of cuts.

Correctness of $PATH1$ follows from the correctness of $FTEST1$, from the fact that all possible candidates for λ^* are included in $\mathcal{M}$, and from the fact that each value discarded is either at most λ_1 or at least λ_2.

Lemma 3.3 cont.: The time for inserting values into R and performing the selections is $O(n)$.

Proof: We give an accounting argument. Charge 2 credits for each value inserted into R. As R changes, we maintain the invariant that the number of credits is twice the size of R, as R changes. When R has k elements, a selection takes $O(k)$ time, paid for by k credits, leaving k credits still available. Then, $k/2$ elements are removed from R, so that the invariant is maintained. Since n elements are inserted into R during the whole of $PATH1$, the time for forming R and performing selections is $O(n)$.

Lemma 3.3 cont.: The number of submatrices inserted into $\mathcal{M}$ is $O(n)$.

Proof: It remains to count the number of submatrices inserted into $\mathcal{M}$. Initially $2n-1$ submatrices are inserted into $\mathcal{M}$. For $j = 1, 2, \ldots, \log n - 1$, consider all submatrices of size $2^j \times 2^j$ that are at some point in $\mathcal{M}$. A matrix that is split must have its smallest value at most λ_1 and its largest value at least λ_2. However, $M_{i,j} > M_{i-k,j+k}$ for $k > 0$, since the path represented by $M_{i-k,j+k}$ is a subpath of the path represented by $M_{i,j}$. Hence, for any submatrix of size $2^j \times 2^j$ which is split, at most one submatrix can be split in each diagonal extending upwards from left to right. There are fewer than $2n$ diagonals, so there will be fewer than $2(n/2^j)$ submatrices that are split. Thus the number resulting from quartering is less than $8(n/2^j)$. Summing over all j gives $O(n)$ submatrices in $\mathcal{M}$ resulting from quartering.