

# USACO JAN07 Problem 'lineupg' Analysis

by Richard Peng

This problem is an instance of the classical RMQ (range minimum/maximum query) problem. It suffices to consider the minimum case of this problem as the maximum case can be done similarly. There are many ways to approach this problem, they are:

1. Iterate through all the elements in the range and determine the minimum. This and other variant (sort all elements, start from each end until elements in the range is encountered) runs in worst case  $O(NQ)$  time and is expected to get 4 out of the 10 test cases.
2. Use a segment tree, which can be implemented as a heap. For each query, combine  $\log N$  segments to find the minimum of the entire segment. This runs in  $O(n \log n)$  time and is expected to get 7~8 test cases.
3. Calculate the minimum/maximum for all segments with lengths being powers of two. Once the minimum of all segments of length  $l$  has been calculated. Let this value be  $M(a, l)$ . Then  $M(a, 2l) = \min(M(a, l), M(a+l, l))$ . It's easy to see this runs in  $O(N)$  time for each length  $l$ . And since  $l$  doubles each time, at most  $\log N$  iterations can be done, which bring the total preprocessing time to  $O(n \log n)$  with  $O(n \log n)$  space usage.

Now to find minimum of a segment, say  $[a, b]$ , we first find the largest power of two which is less than the length of the segment, let it be  $l$ . Then the answer is  $\min(M(a, l), M(b-l+1, l))$ . This is because the two intervals  $[a, a+l-1]$ ,  $[b-l+1, b]$  is guaranteed to cover the entire segment since if we have  $a+l-1$

Overall, this runs in  $O(N \log N + Q)$  time, and is expected to get full points.

code:

```
#include <stdio.h>

int minv[17][50000], maxv[17][50000], n;
FILE *f_in, *f_out;
int mi(int a, int b) {return a < b ? a : b;}
int ma(int a, int b) {return a > b ? a : b;}

main() {
    int i, j, siz, sl, a, b, q;
    f_in = fopen("lineupg.in", "r");
    f_out = fopen("lineupg.out", "w");
    for (fscanf(f_in, "%d%d", &n, &q), i = 0; i < n;
         fscanf(f_in, "%d", &a), minv[0][i] = a, maxv[0][i] = a, i++);
    for (siz = 2, sl = 1, i = 1; (siz <= n); siz *= 2, sl *= 2, i++)
        for (j = 0; j < n; minv[i][j] = mi(minv[i-1][j], minv[i-1][j+sl]),
            maxv[i][j] = ma(maxv[i-1][j], maxv[i-1][j+sl]), j++);
    for (i = 0; i < q; i++) {
        fscanf(f_in, "%d%d", &a, &b);
        a--;
        siz = b - a;
        for (j = 0, sl = 1; siz > 1; sl = sl << 1, j++, siz = siz >> 1);
        fprintf(f_out, "%d\n", ma(maxv[j][a], maxv[j][b-sl]) -
            mi(minv[j][a], minv[j][b-sl]));
    }
    fclose(f_in);
    fclose(f_out);
}
```