Code Jam 2014 - Round 2

Analysis: Up and Down

Given a sequence of distinct numbers, we are asked to rearrange it by using a series of swaps of adjacent elements resulting in an up and down sequence. We want to do it in such a way that the total number of swaps is minimized.

We describe a greedy solution. We use the example sequence 6,5,1,4,2,3 to help explain the solution.

- Repeat N times:
  - Pick the smallest value in the sequence. In the example sequence, it is 1.
  - Determine if the smallest value is closer to the left end of the sequence or the right end of the sequence. In the example, the left end is where 6 is and the right end is where 3 is. The smallest value, 1 is closer to the left end (2 swaps away) than the right end (3 swaps away).
  - Move the smallest value towards the closest end by swapping with adjacent elements, and record the number of swaps. In the example, we move 1 towards 6, first swapping 1 and 5, then swapping 1 and 6 resulting in the sequence: 1,6,5,4,2,3. We performed 2 swaps.
  - Remove the smallest number from the sequence. In the example, we remove 1 resulting in the sequence 6,5,4,2,3.
  - Repeat the process on the resulting sequence. In our example, we repeat the process on sequence 6,5,4,2,3. We identify the smallest element, which is 2, then move it towards the closer end which is the right end with 1 swap, then remove 2 resulting in sequence 6,5,4,3. We then repeat the process on this sequence. First pick smallest value which is 3. Since 3 is already on the right most end, we perform 0 swaps to move it to the closer end, then we remove 3. We repeat a similar process for values 4, 5, then finally 6.
- Finally, report the total number of swaps. In our example, we performed 2 + 1 = 3 swaps.

Thatâ€™s it! You might be wondering why the greedy algorithm works. We provide an intuition here. At each step in our algorithm, we pick the smallest number. In the final up and down sequence, this smallest number will need to go to one of the ends at some point in time. We move this smallest number to the closer end to minimize the number of swaps. After this smallest number reaches the closer end, its position is final, i.e. its position in the final up and down sequence is set, therefore it will never need to be moved/swapped further. Thus, in the next step we can remove that smallest number from consideration and work on a completely new sequence that is one size smaller. We repeat the process for this new sequence, i.e. move the smallest element to the closer end, then remove that number, until we reach the empty sequence. Since we perform the minimum number of swaps in each step on each sub-problem (sequence which is 1 size smaller), we minimize the number of swaps for the whole problem.