SOLUTION TO AN OLD GOOGLE INTERVIEW QUESTION INVOLVING COLORED MARBLES ON A LINE

REMY OCHEI

Abstract. This is a problem adapted from an old Google Interview question I encountered years ago: Given a bucket filled with N different colors of marbles, we wish to lay out the marbles on a line so that marbles of the same color are as far apart from each other as possible. Here we provide at once an algorithm for this problem, along with a proof of correctness.

1. The Problem

Given a bucket filled with N different colors of marbles, we wish to lay out the marbles on a line so that marbles of the same color are as far apart from each other as possible.

2. Ambiguities

Now, there is some ambiguity as to what we are setting out maximize. For instance, we could care about maximizing the minimum distance between any two marbles of the same color or the the average distance between marbels of the same color. It turns out that neither is the sense in which this problem was intended and indeed, both of the two interpretations thusfar offered harbor further ambiguities if adopted.

3. Objective Function

Instead, we wish to maximize the sum of the distances between all pairs of same color marbles. Formally, if M is the total number of marbles, we wish to partition the integers $1, 2, 3, \ldots M$ into N distinct sets such that the sum of the differences of all distinct pairs with each set is maximized across all sets.

Let P_{ij} be the j^{th} position that contains a marble of color C_i. Then, our objective is

$$\max_{P} \sum_{i=1}^{N} \sum_{j=1}^{M_{i-1}} \sum_{k=j+1}^{M_i} \mid P_{ik} - P_{ij} \mid$$

If we assume the members of P_i are sorted, then the absolute value bars are no longer needed. This objective function can be converted into a form that makes the solution readily apparent. Suppose color 1 is red, and suppose there are x red marbles. The highest position that contains a red marble P_{1x} will appear in the objective function $x-1$ times, as there are $x-1$ pairs where P_{1x} is the highest element. In no case is red marble x the lower element of a pair. For red marble $x-1$, there are $x-2$ pairs where it is the higher element and 1 where it is the lower element of the pair. The objective function becomes

$$\max_P \sum_{i=1}^{N} \sum_{j=1}^{M_i} (j-1) \times P_{ij} - (M_i - j) \times P_{ij}$$

or

$$\max_P \sum_{i=1}^{N} \sum_{j=1}^{M_i} (2j - M_i - 1) \times P_{ij}$$

Now, let weight $w_{ij} = (2j - M_i - 1)$. We want a mapping between the integers $1, 2, \ldots M$ and these weights that maximizes the objective function.

Lemma. *If there are two sets of numbers (of equal size), where every member of one set is to be multiplied by a unique member of the other set, the matching that maximizes the sum of products is the matching that pairs elements by rank.*

Proof. For two sets of cardinality two, $\{x, y\}, \{w, z\}$ we prove $xw + yz > xz + yw$,

(1) $x < y$ and $w < z$ (by assumption)

(2) $xz + yw = zy - z(y - x) + wx + w(y - x)$

(3) $xz + yw = zy + wx + (w - z)(y - x)$

(4) $xz + yw = xw + yz - n$ $(n \in \mathbb{Z}+)$

Since by assumption $x < y$, the quantity $y - x$ is positive. Similarly, since $w < z$, $w - z$ is negative. Thus, the entire quantity $(w - z)(y - x)$ is negative and $xz + yw < zy + wx$.
The general cardinality case, via contradiction:
Assume the highest sum of products is achieved via a matching that does not pair the elements by rank. Let y be the greatest element of both sets that is not paired with the coresponding element of equal rank from the other set. Let z be the element of the other set of equal rank to y. Let x be the element paired with z and let w be the element paired with y. This situation is illustrated in figure 1. Then we have,

(5) $x < y$ (since y was taken to be the greatest element not paired by rank)

(6) $w < z$ (since z is the element of equal rank to y from the other set)

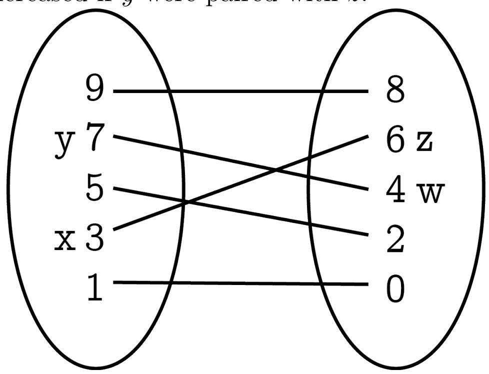
Thus the assumptions for the two element case hold here and the sum achieved can be increased if y and z are paired and x and w are paired □

Therefore, we should order the marbles by their weights $w_{ij} = 2j - M_i - 1$.

4. Example of the algorithm

Input: 3 red, 4 blue, 4 green marbles.
Possible Marble positions: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11.

FIGURE 1. A matching between two sets of numbers that does not respect rank. The sum of products produced by this matching would be increased if y were paired with z.



Weights:

$$\begin{aligned}
w_{11} &= 2(1) - 4 = -2 && C_1 \text{ Red} \\
w_{12} &= 2(2) - 4 = 0 && \\
w_{13} &= 2(3) - 4 = 2 && \\
w_{21} &= 2(1) - 4 = -3 && C_2 \text{ Blue} \\
w_{22} &= 2(2) - 4 = -1 && \\
w_{23} &= 2(3) - 4 = 1 && \\
w_{24} &= 2(4) - 4 = 3 && \\
w_{31} &= 2(1) - 4 = -3 && C_3 \text{ Green} \\
w_{32} &= 2(2) - 4 = -1 && \\
w_{33} &= 2(3) - 4 = 1 && \\
w_{34} &= 2(4) - 4 = 3 &&
\end{aligned}$$

Then, one valid ordering of the weights is $[w_{21}, w_{31}, w_{11}, w_{22}, w_{32}, w_{12}, w_{23}, w_{33}, w_{13}, w_{24}, w_{34}]$ and the output that corresponds with this ordering is [blue, green, red, blue, green, red, blue, green].

$$P = \begin{bmatrix} 3 & 6 & 9 & \varnothing \\ 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \end{bmatrix}$$

5. Complexity

The run-time complexity of generating the weights is $O(M)$, where M is the number of marbles. The weights are generated in sorted order, and the final step of the algorithm is merging the N sorted arrays (recall N is the number of marble colors) of weights into a single sorted array. This merge process is $O(cN)$ where $c = M/N$. The worst case for the merge step is when $M = N^2$ and $c = N$ giving a complexity of $O(N^2)$. However, the complexity for the entire process of generating the weights and merging the weights into a total ordering is

$$\begin{aligned} O(M) + O(cN) & \\ &= O(M) + O(M/N \times N) \\ &= O(2M) \\ &= O(M) \end{aligned}$$

Of No School Nor Style, 1219 Bethel School CT, Coppell, TX 75019
Email address: do.infinitely@gmail.com