CEMC Math Circles - Grade 11/12
March 31, 2020 - April 6, 2020
Comparison Machine - Solutions

Summary of the Tasks

Develop algorithms to complete tasks involving the relative order of n distinct integers.

- The integers are random and unknown to you. All you know is that they are named $a_1, a_2, \ldots a_n$.
- For each task, your approach must work no matter what the order of the integers is.
- A helpful machine M is available. The machine knows the relative order of these integers. To use it, you enter the index of two integers into the machine and it will tell you which of the two corresponding integers is larger.
- For each task, there is a limit on the number of times you can use the machine. This limit applies no matter what the relative order of the n integers happens to be.
- Your memory is perfect and you can remember (or record) the result every time you use the machine.

Details of the Three Tasks

n	Task	Limit
7	Determine the largest integer.	6
8	Determine both the largest integer and the smallest integer.	10
8	Determine the second largest integer.	9
4	List the integers from smallest to largest.	5
9	You are given the additional information that • $a_1 < a_2$ and, • $a_3 < a_4 < a_5 < a_6 < a_7 < a_8 < a_9$. List all 9 integers from smallest to largest.	6
5	Determine the median integer.	6

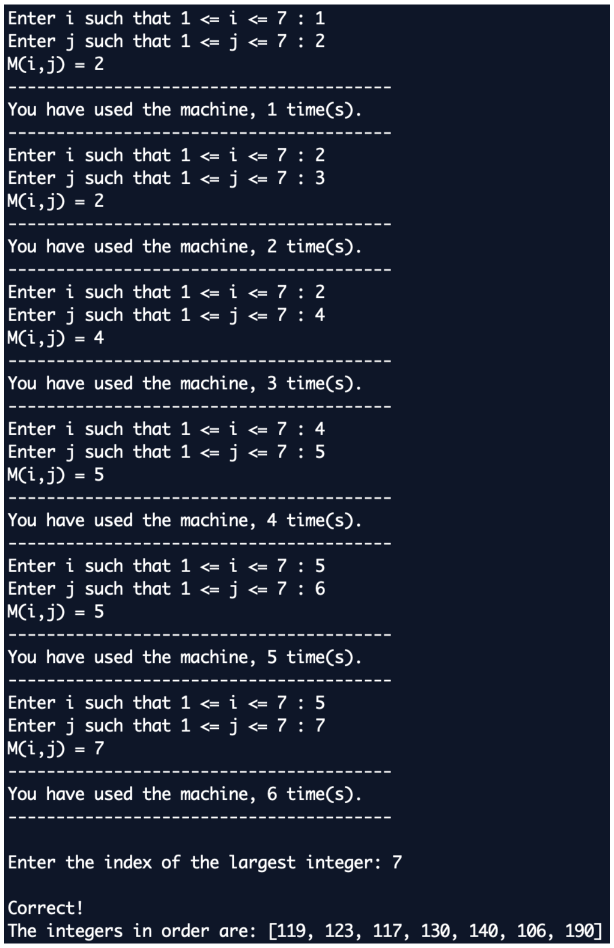
Solutions begin on the next page.

Solution for Task 1

We will keep track of an index named c. The "current maximum" will be at index c. Begin by setting $c = 1$. Then, repeatedly set $c = M(c, i)$ for $i = 2, 3, \ldots, 7$. (Setting $c = M(c, i)$ means we compute the value of $M(c, i)$ and then update c to be this value.) This uses the machine exactly 6 times. We can then declare a_c to be the largest integer because the machine has told us either directly or indirectly that it is larger than each of the other integers.

Can you see why this algorithm works? If the index of the largest integer is 1, then we will see $M(1, i) = 1$ at each stage and the value of c will never change. If the index of the largest integer is not 1, then at some point the value of c will change. In particular, the value of c will change from 1 to i the first time we see a comparison of the form $M(1, i) = i$ for some $i \geq 2$. The value of c will continue to change each time we find an integer that is larger than our "current maximum".

Here is an exampe using the provided tool:



```
Enter i such that 1 <= i <= 7 : 1
Enter j such that 1 <= j <= 7 : 2
M(i,j) = 2
--------------------------------------
You have used the machine, 1 time(s).
--------------------------------------
Enter i such that 1 <= i <= 7 : 2
Enter j such that 1 <= j <= 7 : 3
M(i,j) = 2
--------------------------------------
You have used the machine, 2 time(s).
--------------------------------------
Enter i such that 1 <= i <= 7 : 2
Enter j such that 1 <= j <= 7 : 4
M(i,j) = 4
--------------------------------------
You have used the machine, 3 time(s).
--------------------------------------
Enter i such that 1 <= i <= 7 : 4
Enter j such that 1 <= j <= 7 : 5
M(i,j) = 5
--------------------------------------
You have used the machine, 4 time(s).
--------------------------------------
Enter i such that 1 <= i <= 7 : 5
Enter j such that 1 <= j <= 7 : 6
M(i,j) = 5
--------------------------------------
You have used the machine, 5 time(s).
--------------------------------------
Enter i such that 1 <= i <= 7 : 5
Enter j such that 1 <= j <= 7 : 7
M(i,j) = 7
--------------------------------------
You have used the machine, 6 time(s).
--------------------------------------

Enter the index of the largest integer: 7

Correct!
The integers in order are: [119, 123, 117, 130, 140, 106, 190]
```

The value of c just before each use of the machine and right after the last use of the machine will be

$$1, 2, 2, 4, 5, 5, 7.$$

Solution for Task 2

Begin by computing $M(1, 2)$, $M(3, 4)$, $M(5, 6)$, and $M(7, 8)$ and recording the answers as indices a, b, c, d. Name the other indices e, f, g, h. At this point we have used the machine 4 times and know the largest integer must be at index a, b, c or d. This is because the machine has told us that the integers at indices e, f, g and h are each smaller than at least one of the integers at indices a, b, c or d. Similarly, the smallest integer must be at index e, f, g or h.

Now notice that our solution for Task 1 generalizes to give a way to find the largest of n integers using the machine $n-1$ times. Moreover, if we adjust things by using c to keep track of the index of a "current *minimum*", we can also use it to find the smallest of n integers using the machine $n-1$ times.

Using this generalization, we can find the largest integer among those at indices a, b, c and d using the machine 3 more times. We can also use it to find the smallest integer among those at indices e, f, g and h using the machine 3 more times. These two values are the largest and smallest integers overall and we used the machine $4 + 3 + 3 = 10$ times.

Solution for Task 3

Begin by using the machine 4 times as in the solution for Task 2, and defining a, b, c, and d in the same way. Then compute $M(a, b)$ and $M(c, d)$, recording the answers as the indices x and y. Next, compute $M(x, y)$ and record the answer as the index z. You can think of this approach as a standard "tournament" or "bracket" in a competition where only the winners move on at each stage.

The integer a_z must be the largest integer because it is the only integer that has not been declared smaller than some other integer by the machine. Moreover, the second largest integer can only have been declared smaller than a_z. So to find the second largest integer, we need only find the largest integer among those at indices entered into the machine with z. Note that z was entered into the machine 3 times so we can use our generalized solution to Task 1 by using the machine 2 more times to determine the second largest integer. In total, we used the machine $4 + 2 + 1 + 2 = 9$ times.

Note

We have only shown that we can complete these tasks within the limits of 6, 10 and 9. Do we need the limits to be this high or can they be lowered? It turns out that they cannot be lowered; they are optimal. That is, for each task, there do not exist correct algorithms that always use the machine strictly fewer times than the given limit. Proving this is very challenging.

Solution for Task 4

Compute $M(1, 2)$ and record the answer as a. If $a = 1$, then set $b = 2$. Otherwise set $b = 1$.
That is, we set a equal to the index of the larger integer (or "winner") and set b equal to the index of the smaller integer (or "loser").
Compute $M(3, 4)$ and record the answer as c. If $c = 3$, then set $d = 4$. Otherwise set $d = 3$.
Compute $M(a, c)$ and record the answer as e. If $e = a$, then set $f = c$. Otherwise set $f = a$.
Compute $M(b, d)$ and record the answer as g. If $g = b$, then set $h = d$. Otherwise set $h = b$.
At this point we have used the machine 4 times. We know the integer at index e is the largest overall because it is the "winner of the winners". We know the integer at index h is the smallest overall because it is the "loser of the losers". Finally, compute $M(f, g)$ to determine the order of the other two integers. This process allows us to list all four integers from smallest to largest and we have used the machine 5 times.

Solution for Task 5

First we find the correct place for a_1 among the items in the ordered list $a_3, a_4, a_5, a_6, a_7, a_8, a_9$. That is, we find an index i such that $a_i < a_1 < a_{i+1}$. To do this, we compute M(1,6). If we learn that $a_1 > a_6$, then we compute $M(1, 8)$. Otherwise, we compute $M(1, 4)$. Now we have used the machine twice and we know one of following four things is true (and we know which one is true):

- $a_1 > a_6$ and $a_1 > a_8$, or
- $a_1 > a_6$ and $a_1 < a_8$, or

- $a_1 < a_6$ and $a_1 > a_4$, or
- $a_1 < a_6$ and $a_1 < a_4$.

If the first case above is true, then we know that either $a_8 < a_1 < a_9$ or $a_1 > a_9$. We next compute $M(1, 9)$ to determine which of these is true. In particular, we determine which of $a_3, \ldots, a_8, a_9, a_1$ or $a_3, \ldots, a_8, a_1, a_9$ is a list of integers from smallest to largest. The other three cases shown above are similar. In summary, we can place a_1 correctly using the machine 3 times. Now we simply repeat this process to find the correct place for a_2 in the ordered list $a_3, a_4, a_5, a_6, a_7, a_8, a_9$. Finally, since we know that $a_1 < a_2$, we can list all 9 integers from smallest to largest and we have used the machine $3 + 3 = 6$ times.

(Note that the process of placing a_2 can use the machine less than 6 times in some cases by only placing it within the integers among $a_3, a_4, a_5, a_6, a_7, a_8$ and a_9 that are greater than a_1. However, in the worst-case, we will still need to use the machine 6 times.)

Part of this algorithm is well-known and very important. It is called binary search. Its key idea is to find the correct place for an item in an ordered list by repeatedly comparing it to a "middle item".

Solution for Task 6

A critical piece of this solution is to note that we can use the first 3 comparisons of the solution to Task 1 in order to find the largest of any four integers. Specifically, this involves comparing the integers in pairs and then determining the "winner of the winners". We will call this our *subroutine.*

We begin by using our subroutine with any four of the five integers. The largest of these four integers cannot be the median of the five integers (because it is larger than at least three others) so we discard it. We now need to find the second largest of the remaining four integers since this integer must be the median integer.

Use the subroutine with these four integers. However, from our first use of the subroutine, we already know how two of these integers compare so we can reuse this comparison as our first comparison in the second use of the subroutine. So to complete the second use of the subroutine we only need to do 2 new comparisons. We can discard the largest of the remaining four integers as before. Three integers remain. The largest of these three integers must be the median of the original five integers.

As before, from our previous use of the subroutine, we already know how two of the three remaining integers compare. So we compare the third integer with the larger of these two integers to determine the largest of these three integers. So we have completed our task using the machine $3 + 2 + 1 = 6$ times.