Copyright Notice

The manuscript was published as

> **Acta Informatica 13** (1980): 1–8.

Some beautiful arguments using mathematical induction.

by

Edsger W.Dijkstra

Summary. Three elegant proofs and an efficient algorithm are derived. The derivations evolve smoothly from the choice to apply mathematical induction, the pattern of reasoning that has been chosen as the "Leitmotif" for this small collection. The last proof is the by-product of the algorithm.

Author's Address: Burroughs Corporation
Plataanstraat 5
5671 AL NUENEN
The Netherlands

(To be submitted to ACTA INFORMATICA.)

29 December 1978

Some beautiful arguments using mathematical induction.

The purpose of this article is to exemplify the possibly great role in our reasoning played by mathematical induction. Mathematical induction is of special significance for computing science because the latter deals almost exclusively with a discrete universe of discourse. Furthermore, when applied well, mathematical induction can lead to very compact and effective --in short: beautiful!-- arguments, and when we recognize the battle against chaos, mess, and unmastered complexity as one of computing science's major callings, we must admit that "Beauty is our Business". Finally I hope to demonstrate how a conscious effort at applying mathematical induction can be of great heuristic value.

For educational reasons I have chosen three examples from rather different areas; they are of additional educational value, because the arguments can --and will-- be presented in a way that mirrors closely the way in which they have been discovered (or, if you prefer: have been designed). For none of the results obtained, however, is novelty claimed.

First example.

Let n be a natural number $(n \geq 0)$ and let p be a prime $(p \geq 2)$. Let s be the sum of the p-ary digits needed to represent n in base p; let m be the multiplicity of the factor p in $n!$ --i.e. the maximum value m such that p^m divides $n!$ --. Prove that

$$m = \frac{n - s}{p - 1} \quad . \qquad (1)$$

Because the factorial function is defined recursively by

$$0! = 1 \qquad \text{and} \qquad (n+1)! = (n+1) * n! \quad ,$$

mathematical induction over the natural numbers seems indicated. Expressing the dependence on n therefore explicitly, we rewrite (1) as

$$m(n!) = \frac{n - s(n)}{p - 1} \qquad (2)$$

and look for a base and an induction step, valid for any prime p.

The base is easy: because $m(0!) = m(1) = 0$ and $s(0) = 0$, relation (2) holds for $n = 0$.

For the induction step we observe first that, because p is prime,

$$m((n+1)!) = m(n!) + m(n+1) \quad ,$$

i.e. replacing n by $n+1$ increases the left-hand side of (2) by $m(n+1)$.

Realizing that $m(n+1)$ equals the number of zero digits with which the p-ary representation of $n+1$ ends, and therefore also equals the number of p-1 digits with which the p-ary representation of n ends --performing the subtraction of 1 in base p makes that last conclusion obvious!-- we deduce

$$s(n+1) = s(n) + 1 - (p-1) * m(n+1) \quad ,$$

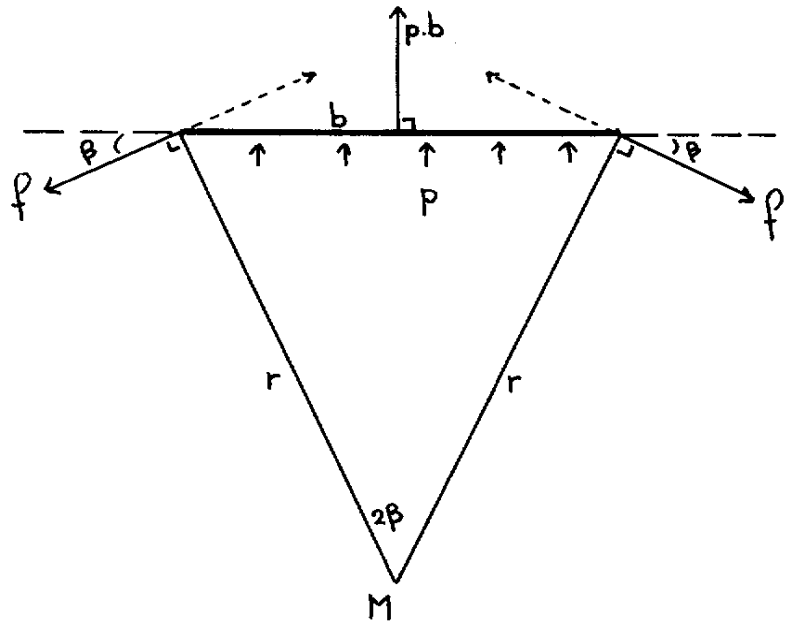
from which it follows immediately that replacing n by $n+1$ increases also the right-hand side of (2) by $m(n+1)$. Q.E.D.

Second example.

Here we deal with the problem, what shape a polygon with sides of given --possibly different-- lengths will take if the enclosed area has to be of maximum value. Because the number of sides may be any number ≥ 3, an inductive argument dealing with the sides, for instance in the order in which they occur along the circumference, would be nice. But because nothing has been given about the lengths, an induction step that is independent of the lengths of the sides would be the nicest of all.

With the polygon built from rigid sides, flexibly joined in their end points to their two neighbours, the shape enclosing the maximum area is stable when a constant outward pressure p is exerted on all sides. (This is because

under changing shape the pressure performs an amount of work equal to p * the increase of the area enclosed.) Hence the outward force on each side, that results from the pressure p , is entirely compensated by the forces of reaction exerted upon it by its two neighbours.



The outward force resulting from the pressure p exerted on a side of length b equals p.b and is applied to its midpoint in a direction orthogonal to it. The stability of the side requires that the parallel components of the two reaction forces cancel, and that their orthogonal components be p.b/2 . Hence the two reaction forces have the same value, f say, and make the same angle, β say, with the side, the four quantities being related by

$$p.b = 2.f.\sin(\beta) \quad .$$

From each end point we now draw a line orthogonal to the force of reaction, and call the point where these two lines meet M . Because the two lines meet at an angle $2.\beta$, and the point M lies at the same distance, r say, from the two endpoints, we have

$$b = 2.r.\sin(\beta) \quad .$$

Combining the above two relations we derive

$$r = f/p \quad .$$

Knowing the physical law "action = reaction" and also the fact that the pressure on the next side has to be compensated by two reaction forces of the same value, we conclude that the pressure on the next side is compensated by two reaction forces that also have that same value f. The pressure p being constant as well, the top of the isosceles triangle, similarly constructed for the next side, has therefore the same distance r from the endpoints of that next side. Furthermore, because in each joint the forces of reaction have opposite directions, the orthogonals to them coincide. Combining the two we conclude that the top of the isosceles triangle constructed on the next side coincides with M. Hence, without any assumption about the length of the next side we have derived that also its other endpoint lies at a distance r from the point M. Via mathematical induction along the circumference we may now conclude that all vertices of the polygon have the same distance r from the point M. Hence we have proved that the vertices of the polygon with given lengths of its sides and with maximum enclosed area lie on a circle.

The interchange of two neighbouring sides leaves the area enclosed and the radius of the circumscribed circle unchanged. Because any permutation can be written as the product of interchanges of pairs of neighbours, we conclude, again via mathematical induction, that the maximum area enclosed and the radius of the circumscribed circle depend only on the lengths of the sides, but not on the order in which these lengths occur along the circumference.

Third example.

For $N \geq 1$ we consider a sequence of N elements: $A[0],\ldots,A[N-1]$. The order of increasing subscript value will be called "the order from left to right". For any s satisfying $0 \leq s \leq N$, we can take from $A[0],\ldots,A[N-1]$ so-called "subsequences of length s" by removing an arbitrary collection of $N-s$ elements and retaining the remaining s elements in the order in which they occurred in the original sequence. As a result, $A[0],\ldots,A[N-1]$ contains 2^N subsequences. When, in addition, each element has an integer value, we call a subsequence an "upsequence" if and only if it contains no element with a right-hand neighbour smaller than itself.

Note. According to this definition, all N subsequences of length 1 --and even the empty subsequence-- are upsequences. (End of note.)

Our problem is the design of an algorithm that determines for any such sequence the maximum length of an upsequence contained in it.

Note. Although there need not be a unique longest upsequence, the maximum length is unique, e.g. the given sequence $(3,1,1,2,5,3)$ yields 4 for the maximum length, realised either by $(1,1,2,5)$ or by $(1,1,2,3)$. (End of note.)

Let the final value of the variable k represent the answer we are looking for, i.e. we seek to establish the relation

R: k = the maximum length of an upsequence contained in $A[0],\ldots,A[N-1]$.

A moment's reflection tells us that each element of the given sequence has to be considered, and we only make the (modest) assumption that we can get away with taking the elements into consideration in the order from left to right. More formally, we propose to introduce a second variable, n say,

and to establish initially and to maintain subsequently the so-called "invariant relation"

P1: k = the maximum length of an upsequence contained in $A[0],\ldots,A[n-1]$ and $1 \leq n \leq N$,

to be used in a program of the structure --assertions having been inserted between braces--

"establish P1 for $n = 1$ "; $\{P1\}$
do $n \neq N \rightarrow \{P1$ and $1 \leq n < N\}$
"increase n by 1 under invariance of P1 " $\{P1\}$
od $\{P1$ and $n = N\}$.

Relation P1 has been inspired by the fact that R contains the parameter N ; it has been derived from R by the standard technique of replacing a constant (here N) by a variable (here n) --and (as usual) restricting its range-- so that, as a result

$$(P1 \text{ and } n = N) \Rightarrow R \quad ,$$

from which we deduce that the above program would do the job; this inspiration is further encouraged by the observation that P1 is easily established initially as $(n = 1$ and $k = 1) \Rightarrow P1$.

The repeatable statement "increase n by 1 under invariance of P1 " is the algorithmic equivalent of the induction step: given the solution for n it has to construct the solution for n+1 . The required invariance of P1 means that the increase $n := n + 1$ may have to be accompanied by an adjustment of the value of k (the only other variable occurring in P1 !). Because extension of the sequence considered with a next element can never decrease the maximum length of an upsequence contained in it, and can increase it by at most 1 , the adjustment of k , when needed, will have the form $k := k + 1$. For the repeatable statement we can take the form

```
{P1 and 1 ≤ n < N}
if .... → k:= k + 1 [] .... → skip fi;
n:= n + 1 {P1}
```

and our only task is now to fill in the dots, i.e. to decide under which circumstances k has to be increased by 1, or can remain unchanged respectively, such that after the subsequent increase $n:= n + 1$ the relation $P1$ is again guaranteed to hold.

Because $A[n]$ is the next element to be considered, we can fill in the dots as follows:

```
{P1 and 1 ≤ n < N}
if m ≤ A[n] → k:= k + 1 [] A[n] < m → skip fi;
n:= n + 1 {P1}
```

provided the value of m is defined by

D: m = the minimum right-most element of an upsequence of length k contained in $A[0],...,A[n-1]$.

In other words: the obligation to keep $P1$ invariant requires, besides the value k, the value m as an additional derivative from $A[0],...,A[n-1]$. Introducing m as a variable, and replacing the original invariant relation $P1$ by the stronger $P1$ and D, we find ourselves considering the program

```
n:= 1; k:= 1; m:= A[0]; {P1 and D}
do n ≠ N → {P1 and D and 1 ≤ n < N}
           if m ≤ A[n] → k:= k + 1; m:= A[n]
            [] A[n] < m → ....
           fi;
           n:= n + 1 {P1 and D}
od {R}
```

Note. Strengthening the invariant relation is the computational analogue to the strengthening of an induction hypothesis. (End of note.)

Note that now we have to increase n by 1 under invariance of

P1 and D . In the first alternative the invariance of D presents no problem: all upsequences of the increased length have A[n] as their right-most element, and this is therefore the proper new value for m .

But what in the second alternative, when $A[n] < m$? The new element A[n] cannot be used to form a longer upsequence, but should it be used to lower m because, thanks to its inclusion, the right-most element of an upsequence of length k can now be smaller than was possible before the extension? A moment's reflection will tell us that for our last dots we can fill in

$\{A[n] < m\}$
if $m' \leq A[n] \rightarrow m := A[n]$
[] $A[n] < m' \rightarrow$ skip
fi

provided the value of m' is defined by

D': $m' =$ if $k = 1 \rightarrow$ minus infinity
[] $k > 1 \rightarrow$ the minimum right-most element of an upsequence of length k-1 contained in A[0],...,A[n-1]
fi .

In other words: the obligation to keep D invariant requires, besides the values k and m , the value m' as an additional derivative from A[0],...,A[n-1] . After the introduction of m' as a variable and of the strengthened relation P1 and D and D' , the obligation to maintain the invariance of D' requires an m" , etc., and by mathematical induction we conclude that we could use a whole array of m-values, and combine D and D' and D" and ... into

P2: $(A\ j: 1 \leq j \leq k:$ m[j] = the minimum right-most element of an upsequence of length j contained in A[0],...,A[n-1]$)$

where the old m is now m[k] , the old m' is now m[k-1] , etc.

With the new invariant relation P1 and P2 our program takes its

final shape:

```
n:= 1; k:= 1; m[1]:= A[0]; {P1 and P2}
do n ≠ N → "increase  n  by  1  under invariance of  P1 and P2" od {R}
```

We leave to the reader the now straightforward verification that for the repeatable statement the following suffices:

```
"increase  n  by  1  under invariance of  P1 and P2":
if m[k] ≤ A[n] → k:= k + 1; m[k]:= A[n]
 [] A[n] < m[1] → m[1]:= A[n]
 [] m[1] ≤ A[n] < m[k] → "establish  j  such that  m[j-1] ≤ A[n] < m[j]";
                         m[j]:= A[n]
fi;
n:= n + 1 {P1 and P2}
```

Using

P3: $m[i] \leq A[n] < m[j]$ and $1 \leq i < j \leq k$

as the invariant relation in the binary search for our last refinement

```
"establish  j  such that  m[j-1] ≤ A[n] < m[j]":
{m[1] ≤ A[n] < m[k]}
i:= 1; j:= k; {P3}
do i ≠ j - 1 → h:= (i + j) div 2; {i < h < j}
                if m[h] ≤ A[n] → i:= h {P3}
                 [] A[n] < m[h] → j:= h {P3}
                fi {P3}
od {m[j-1] ≤ A[n] < m[j]}
```

we have solved our original problem with an N.log(N)-algorithm.

Note. Because the difference $j - i$ decreases each time by at least 1 and remains positive, termination of our last refinement is guaranteed, and the existence of a j such that $m[j-1] \leq A[n] < m[j]$ is thereby proved. (End of note.)

If we assume the elements of the array m initialized to plus infinity, we observe that in the above algorithm the adjustment of the array m , prior to the increase n:= n + 1 , boils down to the effective decrease of exactly one element of the array m to the value A[n].

Suppose that we determine in parallel h = the maximum length of a downsequence contained in A[0],...,A[N-1] . That computation would comprise a corresponding array, p say, --with elements initialized to minus infinity-- such that each adjustment of it boils down to an effective increase of exactly one element of the array p to the value A[n].

We call a pair (i, j) --with $1 \leq i \leq k$ and $1 \leq j \leq h$ -- such that $m[i] \leq p[j]$ "an inversion". Because m-values never increase and p-values never decrease, an inversion, once introduced, remains in existence. Furthermore the double adjustment introduces at least one new inversion (via the m- and p-elements that are effectively decreased and increased to A[n] respectively). Hence for the combined computation the relation

$$n \leq \text{the number of inversions}$$

is an invariant. Because by definition

$$\text{the number of inversions} \leq h * k$$

we conclude $n \leq h * k$. Hence we have proved the

Theorem. A sequence of length $> M^2$ contains a monotonic subsequence of length $> M$.

And this concludes my treatment of the third example.

(Note added during revision. In reponse to the original manuscript, J.Misra of the University of Texas at Austin, Texas, U.S.A., communicated to me a very nice proof of the last theorem that was based on the Pidgeonhole Principle.)

Plataanstraat 5
5671 AL NUENEN
The Netherlands

prof.dr.Edsger W.Dijkstra
Burroughs Research Fellow