

## Induction solutions

---

### 1. [Grasshopper induction]

This is a definition of basic induction - an induction axiom, if you will. It is based on a notion that all natural numbers are enumerable.

### 2. [Grasshopper induction]

This is a definition of extended induction - our induction step proof relies not only on previous  $k$  but on all  $k$  from base up.

### 3. [Triangular numbers]

Use induction.

- Base  $k = 1$ :

$$T_1 = \frac{1(1+2)}{2} = \frac{2}{2} = 1 \quad - \text{True!}$$

- Assumption (Grasshopper kind) - formula holds for  $k$  -

$$T_k = \frac{k(k+1)}{2}$$

- Proof of induction step:

$$T_{k+1} \stackrel{?}{=} \frac{(k+1)(k+2)}{2}$$

We notice that every  $T_n$  is a triangle whose base length is  $n$ , and that, if we remove this bottom row, we are left with a smaller triangle  $T_{n-1}$ , i.e.  $T_n = n + T_{n-1}$ . Therefore

$$T_{k+1} = T_k + (k+1)$$

Now we substitute  $T_k$  from our assumption and get:

$$T_{k+1} = \frac{k(k+1)}{2} + k+1 = \frac{k(k+1) + 2(k+1)}{2} = \frac{(k+1)(k+2)}{2}$$

and we are done!

### 4. [Waterworks] [http://problems.ru/view\\_problem\\_details\\_new.php?id=35714](http://problems.ru/view_problem_details_new.php?id=35714)

We will prove the more general problem - Peter can equalize any  $2^n$  glasses - and 64 will be a particular case for it.

Use induction on  $n$ .

- Base  $n = 1$ . We have a table with just  $2^1 = 2$  glasses and one move reaches our goal.
- Assumption - Peter knows how to do it for  $2^{n-1}$  glasses (lets name these  $A_1, A_2, \dots$ ).
- Proof of step (can it be done for  $2^n$  glasses?).

We start by splitting up all  $2^n$  glasses into  $2^{n-1}$  pairs  $(A_1, B_1); (A_2, B_2); \dots; (A_{2^{n-1}}, B_{2^{n-1}})$ . We equalize the amount of water in each pair (using  $2^{n-1}$  moves, but who is counting?). Notice that after this equalization the sums of amounts will be the same  $S = A_1 + A_2 + \dots + A_{2^{n-1}} = B_1 + B_2 + \dots + B_{2^{n-1}}$  (because  $A_i = B_i$ )

## Induction solutions

---

Now, Peter takes first glass of each pair -  $A_1, A_2, \dots, A_{2^{n-1}}$  - and equalizes them, getting  $A'_1, A'_2, \dots, A'_{2^{n-1}}$  (he knows how, by assumption, because there are  $2^{n-1}$  of them). Notice that still  $A'_1 + A'_2 + \dots + A'_{2^{n-1}} = S$ , because Peter was not touching  $B$  glasses. Therefore  $A'_i = \frac{S}{2^{n-1}}$

After he is done, he takes all the second glasses from each pair -  $B_1, B_2, \dots, B_{2^{n-1}}$  - and does the same to them, getting  $B'_i = \frac{S}{2^{n-1}}$

Now all the glasses are equal and we are done!

5. [Onewayland] [http://problems.ru/view\\_problem\\_details\\_new.php?id=30825](http://problems.ru/view_problem_details_new.php?id=30825)

Use induction on number of towns.

- Base  $k = 2$ . With two towns it is obvious.  $k = 1$  feels less intuitive, but it also holds because you can always reach a town from itself.
- Assumption - statement holds for any configuration of  $k$  towns and roads between them.
- Proof of step. Say we have a  $k + 1$  town Onewayland. We choose one of the towns -  $H$  - and hide it along with all the roads connecting it. We are left with a  $k$  town Onewayland, in which, by our assumption, there exists a town  $A$  from which all the other towns in this reduced Onewayland are reachable.

Now let's unhide our town  $H$ . If any road goes into  $H$ , then  $H$  is reachable from  $A$  and  $A$  is the town we are looking for in our full Onewayland. If no road goes into  $H$  (but all towns are connected by roads) then a road must go from  $H$  to  $A$  and that means that any other town is reachable from  $H$  - just go to  $A$  first.

6. [Another Onewayland] [http://problems.ru/view\\_problem\\_details\\_new.php?id=30824](http://problems.ru/view_problem_details_new.php?id=30824)

Statement in graph language - if in a connected directed graph every vertex has same number of ingoing as outgoing edges, the graph has an Euler cycle.

Use caterpillar induction on  $n$  - number of edges in a graph.

- Base  $n = 0$ . Only connected graph with 0 edges is a one vertex graph and there statement holds.
- Assumption: statement holds for all graphs with number of edges less than  $k$  (plus the original restrictions)
- Proof of step:

Notice, that if, in a connected directed graph, every vertex has an equal number of in- and out-edges then there is a directed cycle (proof is almost the same as in Graph 1 [Forestry] problem, hint: assume opposite, look at end of longest path).

We find such a directed cycle and erase the edges (put into our stack of final path). Graph is now split into several parts, but each part has all the original conditions (since we removed exactly 1 in- and 1 out- edge from some of the vertices) and number of edges in each part is  $< k$ .

Therefore, for each of the parts we find the Eulerian cycle (by assumption it can be done) and then re-combine them in order of touchpoints in the original cycle, connected by edges in original cycle.

7. [One more Onewayland] [http://problems.ru/view\\_problem\\_details\\_new.php?id=30827](http://problems.ru/view_problem_details_new.php?id=30827)

We will use induction on number of towns  $n$ .

## Induction solutions

---

- Base  $n = 3$ . It easy to see that for 3 vertex graph it holds (it is interesting case since base has to be greater than 1 - it obviously does not work for  $n = 2$ ).
- Assumption: Property holds for all graphs with size  $n - 1$
- Proof of step:

Note that in any full directed graph there is no more than one vertice that has all inbound edges and no more than one vertice with all outbound edges.

We now find and remove a vertice  $v$  that has both in- and out- bound edges (it exists, since no more than 2 vertices does not fit this requirement). Mark two of those vertices  $v_{inbound}$  and  $v_{outbound}$ .

We hide  $v$  and its edges. Remaining graph is a  $n - 1$  vertice graph and for it the property holds by assumption.

Now unhide  $v$  and notice that property still holds, because it is always possible to travel from random  $v_i$  to  $v$  by going first to  $v_{inbound}$  and then to  $v$  and it is possible to travel from  $v$  to random  $v_i$  by going first to  $v_{outbound}$ .

8. [Corner tiling of square] [http://problems.ru/view\\_problem\\_details\\_new.php?id=35522](http://problems.ru/view_problem_details_new.php?id=35522)

We will prove this for any square  $2^n \times 2^n$ .

Use induction on  $n$ .

- Base  $n = 1$ . Cutting out any of  $2 \times 2$  cells leaves us with exactly one "corner".
- Assumption: It holds for  $2^{n-1} \times 2^{n-1}$  square
- Proof of the step:

Say we have a  $2^n \times 2^n$  square with one cell cut out. We can cut this big square into 4 squares  $2^{n-1} \times 2^{n-1}$ .

The missing cell is in one of these squares and, by our assumption, that square can be tiled.

Now, in the each of remaining 3 squares we "paint" the corner cell closest to big squares center. These 3 cells for a "3-cell corner" and the rest of the  $2^{n-1} \times 2^{n-1}$  squares can be tiled by our inductive assumption.

9. [Foxy problem] [http://problems.ru/view\\_problem\\_details\\_new.php?id=65864](http://problems.ru/view_problem_details_new.php?id=65864)

Notice that, if both rabbits have at least 1 berry each, then they will still have at least 1 berry each.

We will now prove that fox can leave only 1 berry to each of  $n$  rabbits. Lets use notation  $\underbrace{(1, 2, 4, \dots, 2^{n-1})}_n$

for distribution of berries to  $n$  rabbits. We will prove that fox can do a series of operations to transform

$$\underbrace{(1, 2, 4, \dots, 2^{n-1})}_n \rightarrow \underbrace{(1, 1, 1, \dots, 1, 2^{n-1})}_n \rightarrow \underbrace{(1, 1, 1, \dots, 1)}_n$$

Use induction on number of rabbits  $n$ .

- Base  $n = 1$ . Conveniently one rabbit already has only one berry, done.
- Assumption: Fox can do

$$\underbrace{(1, 2, 4, \dots, 2^{n-2})}_{n-1} \rightarrow \underbrace{(1, 1, 1, \dots, 1, 2^{n-2})}_{n-1} \rightarrow \underbrace{(1, 1, 1, \dots, 1)}_{n-1} \quad (1)$$

## Induction solutions

---

- Proof of step:

We first look only at  $n - 1$  younger rabbits. By assumption (??), fox can eat all their berries:

$$\underbrace{(1, 2, 4, \dots, 2^{n-2})}_{n-1}, 2^{n-1} \rightarrow \underbrace{(1, 1, 1, \dots, 1, 2^{n-1})}_{n-1}, 2^{n-1} \rightarrow \underbrace{(1, 1, 1, \dots, 1, 1)}_{n-1}, 2^{n-1}$$

Now fox uses her operation on two eldest rabbits

$$(\dots, 1, 2^{n-1}) \rightarrow (\dots, 2^{n-2}, 2^{n-2})$$

.

Then fox, again ignoring the eldest rabbit and using assumption (??), transforms

$$\underbrace{(1, 1, 1, \dots, 1, 2^{n-2})}_{n-1}, 2^{n-2} \rightarrow \underbrace{(1, 1, 1, \dots, 1, 1)}_{n-1}, 2^{n-2}$$

Lastly, ignoring the youngest rabbit and using assumption (??), fox can do

$$(1, \underbrace{1, 1, \dots, 1, 2^{n-2}}_{n-1}) \rightarrow (1, \underbrace{1, 1, 1, \dots, 1}_{n-1})$$

which concludes her nefarious tactics.

Now we know, how much berries remain in rabbits' paws - 100, so, to calculate how much has a fox eaten, we need to calculate the sum  $\sum_{i=1}^{n-1} 2^i$ , which is  $2^n - 1$  (hint - look at binary representations).

So, ultimately, the fox has eaten  $2^n - 101$  berries!