

Old technologies in the development of algorithmic thinking: a Construction Manager game

Alexander K. Zvonkin*

Abstract

We present a simple environment, accessible even for very young children, which permits to develop certain aspects of algorithmic thinking. The activities described here do not necessitate a use of a computer.

The game presented here was first used for very young children (6–7 years old), as it is described in the book [3]. Later on, and in a more developed form, it was used for middle-high school students (12–14 years old). The corresponding series of exercises constitutes a chapter of the manual [4] which was a joint work with Sergei Lando and Alexei Semenov, with a participation of Alexander Kulakov.

1 Presentation of the game

In Fig. 1 you see a “building” made of bricks. The bricks are marked by numbers. You are the Construction Manager. There are several construction teams at your disposal, and you must distribute the work among them.

Let us establish construction rules.

- Any brick (any block), independently of its form and size, can be constructed by a single team in one day.
- Two teams cannot work with one block.
- A block can be put in place only after all blocks it rests on are constructed.

Thus, block 3 can be constructed only after blocks 1 and 2 are constructed. Blocks 3 and 11 provide an example of another kind. They are “independent”, which means that the order they are constructed can be arbitrary: block 3 can be constructed either before block 11, or after it, or even the same day provided we have more than one team. Now several questions may be asked about the building of Fig. 1.

*LaBRI, Université Bordeaux I, 351 cours de la Libération, F-33405 Talence Cedex FRANCE; e-mail: zvonkin@labri.fr

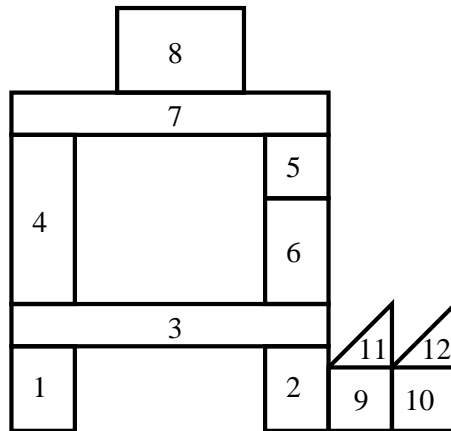


Figure 1: In what order the bricks must be put in their places?

Problem 1. You have only one team. Write a program for constructing the building. Compare it with the program of other students.

In this way the students will see that the solution of the problem is not unique. The presentation of solutions may differ: it may be just a column of block numbers, or a column of “commands” like `construct (1)`, or `put 1`, or something similar. The teacher may discuss the notation with his or her students and they may agree together upon the best way of solution presentation.

Problem 2. The teacher gives the students three different construction plans and says that one of them contains an error. Find the error.

Problem 3. How many days does the execution of each (correct) program take—and why?

Obviously, it takes 12 days, since there are 12 blocks to construct. Now let us slightly change the situation.

Problem 4. You have as many teams as you need. Write a program for constructing the building in the shortest possible time.

Up to now we did not agree about how *simultaneous* work of several teams will be denoted. Let us introduce the following *syntax rule*:

- If several teams work simultaneously, then we write the commands they execute in one row.

Now the solution of Problem 4 looks as follows:

| | | | |
|---|----|----|----|
| 1 | 2 | 9 | 10 |
| 3 | 11 | 12 | |
| 4 | 6 | | |
| 5 | | | |
| 7 | | | |
| 8 | | | |

Each day all the blocks ready to be put in their places are constructed. The plan requires four teams: were there more teams, we would not be able to supply them with work. But the task is fulfilled in six days.

Problem 5. Why can it not be done faster?

The last question necessitates a more profound reflection. Fig. 2 shows what in operations research is called a *critical path*: it is a sequence of six blocks that can be constructed only successively, in a given order. Hence their construction requires at least six working days, even if we disregard other blocks. A critical path is not necessarily unique: indeed, instead of block 1 we could start with block 2.

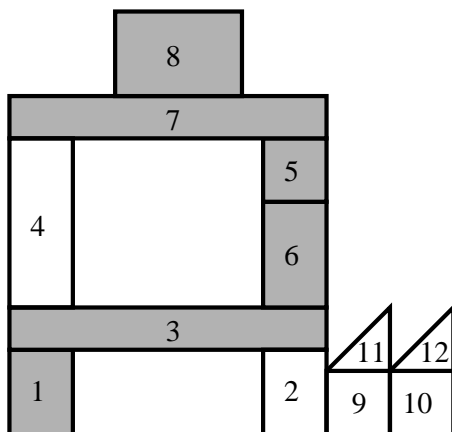


Figure 2: A critical path: the shaded blocks can be constructed only successively.

Having in mind the above observation we may formulate the following

Problem 6. We have to construct 12 blocks, which can be done in at least 6 days. Is it possible to complete the task using only two teams both working each day?

Here is one of the possible programs. In boldface, we show “unavoidable” jobs which cannot be postponed:

| | |
|---|----|
| 1 | 2 |
| 3 | 9 |
| 4 | 6 |
| 5 | 11 |
| 7 | 10 |
| 8 | 12 |

The secret is not to skip blocks in the critical paths. For example, the first day we must construct both blocks 1 and 2 since both lie on critical paths. The second day, block 3 must be constructed. For the other block either of blocks 9 and 10 can be chosen, but the choice of one of them is inevitable since otherwise the second team would be idle! The third day, one must construct block 6; for its pair, either of the blocks 4, 11 and 10 can be chosen. At this moment, construction of block 4 can be postponed, but if it is not constructed after four days, our plans will fail. This means that a program can produce new critical paths itself.

Now let us consider another example, see Fig. 3.

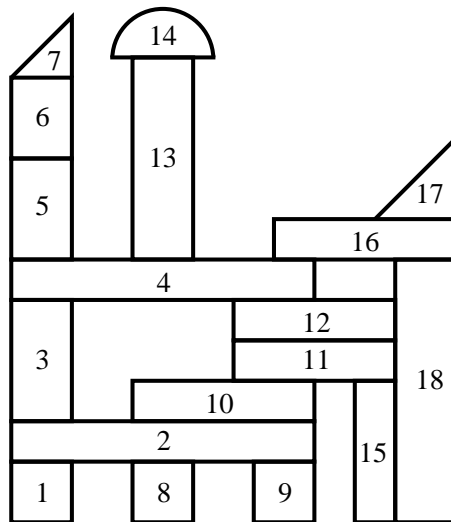


Figure 3: There are 18 blocks to construct, and the shortest time to do that is 9 days. Is it possible to construct the entire building in 9 days by 2 teams?

Problem 7. By finding a critical path of length 9, show that the work cannot be done in less than 9 days.

Problem 8. We have 18 blocks to construct, and 9 days to perform this construction. Is it possible to manage with two teams?

The answer to the last question is negative, but the proof of this fact is rather subtle. Let us discuss it.

- Look at block 2. It must be constructed the second day, otherwise we will be late. But this block is placed on three blocks, namely, 1, 8, and 9; these cannot be constructed during the first working day since we have only two teams. This is the first obstacle, and it leads to a one-day delay. (One may argue that block 2 can be put safely on 1 and 9 only, but it is against the rules.) Note that the second team does not have to be idle the second day like in the previous example: it may build one of the blocks 15 or 18. But at some later stage, it will necessarily be idle.
- Now look at block 4. We would like to construct it the sixth day, since there are three floors above it (blocks 5, 6, and 7). We are one day late already, hence this block will be constructed only on the 7th day instead of the 6th. But after it is constructed there remain 7 blocks yet to be constructed, all those who rest on it. And if we have only two teams, then there is no way to construct 7 blocks in three days. At least four days are necessary.

Problem 9. Construct the building of Fig. 3 by three teams in 9 days. Construct it by two teams in 11 days. Which of the plans, in your opinion, is better?

The notion of a “better” plan is informal: the only purpose of the question is to provoke a discussion among children. One of the possible answers is as follows: choosing the first plan, the manager must pay $3 \cdot 9 = 27$ daily salaries, while the second plan costs only $2 \cdot 11 = 22$ daily salaries. But finishing the work two days earlier the manager may get some profit. If this profit is larger than 5 daily salaries than the first plan is preferable.

Now let us consider one more example (see Fig. 4) since it provides us with yet a new idea.

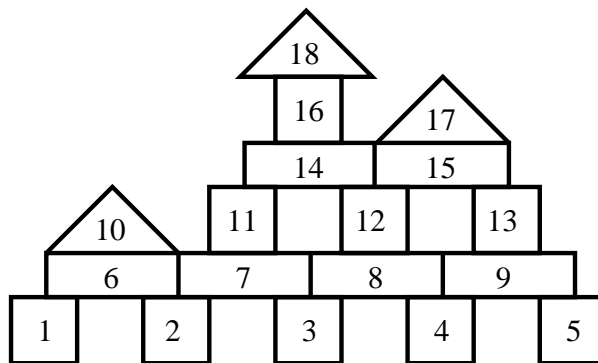


Figure 4: 18 blocks, 3 teams, 6 days: the solution is unique.

Problem 10. Write a program for three teams to construct the building shown in Fig. 4 in 6 days, and *prove that your solution is unique*.

Why is it unique? There are 5 blocks on the first floor, thus $\binom{5}{3} = 10$ possibilities to choose three of them. If we choose to construct, say, blocks 1, 2, and 3 the first day then the choice for the second day is among 4, 5, 6, 7, thus 4 possible choices of three blocks to construct. And if we choose for the first day blocks 1, 3, 5, then the second day one of the teams will be idle. There are too many possibilities to analyse them properly...

What helps here is the analysis from the end to the beginning. The last day the only blocks we can put on are those which do not support any other block. There are exactly three such blocks, namely, the triangle shapes 10, 18, 17. For block 18 this is not surprising since it finishes a critical path and cannot be constructed earlier anyway. But block 17 could in principle be constructed the 5th day, and block 10, even the 3rd day. These are traps to avoid.

Now we take away these three blocks and repeat the same analysis for the 5th day. We find out that the only blocks to be constructed the 5th day are 6, 16, and 15. Continuing the same way, we find that the construction plan is indeed unique. For example, while block 1 can be constructed the 1st day, its construction not only *may* but *must* be postponed until the 4th day!

This principle of “starting from the end” is rather general. It permits to detect the jobs lying on critical paths not only on the first day but at any intermediate moment. We label the “floors” from the top to the bottom; these labels show how many days there remain until the end of the whole construction process “if everything goes well”. Then, whatever is the number of available teams, we must always choose to construct the blocks with bigger labels.

2 Didactical remarks

- The Construction Manager game allows various presentations. For example, playing it with little children (see [3]) we used real wooden blocks of different sizes and shapes. For older children, a more abstract representation is possible. While a computer and an appropriate software might be useful on some occasions, we insist on the fact that the game can be played *without computers*, as a purely paper-and-blackboard work. Hence the phrase “old technologies” in the title of our paper.
- This activity involves not only designing algorithms but also *thinking* about them: proving impossibility or uniqueness, analysing efficiency, and so on. The efficiency criteria may be proposed by students themselves.
- Cooperation of several teams may be considered as a first experience in *parallel programming*. Under certain circumstances it may even be used to enhance motor skills of younger children: “two teams” are two hands, and the child is asked to put two objects in two places simultaneously.

- The most important feature of the Construction Manager environment is its *openness*. Rules may change, though any change of rules should be made very explicit. New problems may be invented by the teacher and by the students themselves. An exchange of problems invented by the students themselves should be encouraged. (A piece of advice to the teacher: from the very beginning, impose an upper bound on the number of blocks, since children have a tendency to invent problems which are too complicated.)
- For further ideas, see the next section.

3 Ramifications of the subject

In this section we give some indications of how the subject may evolve and be enriched.

3.1 PERT charts

Fig. 5 provides another representation of dependencies between the works of Fig. 4. In operations research this graph is called PERT chart (PERT means Program Evaluation and Review Technique), or else a network diagram. In mathematics, it is called a Hasse diagram of a partially ordered set, and a “program for one team” is called a linear extension of the partial order in question. In computer science, algorithms for finding a linear extension of a given partial order are called topological sorting algorithms. For an efficient algorithm of topological sorting see, for example, [1].

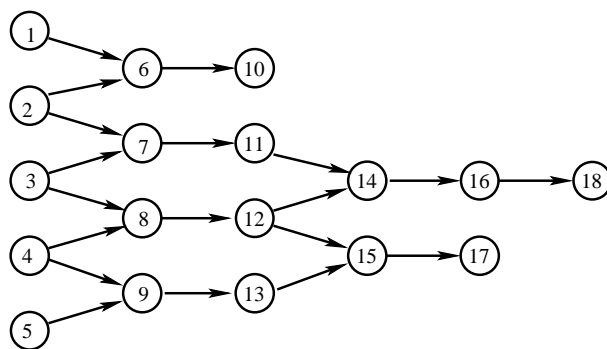


Figure 5: The PERT chart for the building of Fig. 4.

We mention all this terminology to provide an interested teacher with possible hints and sources of new ideas. Let us discuss, for example, the following

Problem 11. Is it possible to realize the chart on the left of Fig. 6 as a task of a building construction?

The start is easy (Fig. 6, middle); but now we are stuck: how to make block 5 dependent only on blocks 1 and 3, without being dependent on block 2? Several fruitless attempts may almost convince us that it is not possible. However, it is—if we admit blocks of fancy shapes (see Fig. 6, right).

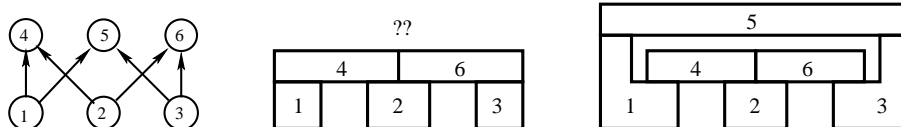


Figure 6: The building on the right represents the chart on the left. In the middle, a dead-locked attempt to solve the exercise.

Problem 12. Is it possible to realize the chart on the left of Fig. 7 as a task of a building construction?

This time the situation is more subtle. In our previous presentation, there was an implicit convention: all our buildings were two-dimensional, and a block x was “immediately dependent” on block y if x touched y . But the graph of Fig. 7 is not planar; therefore, it cannot be realized by a two-dimensional figure. In three dimensions the realization is simple: see Fig. 7, right.

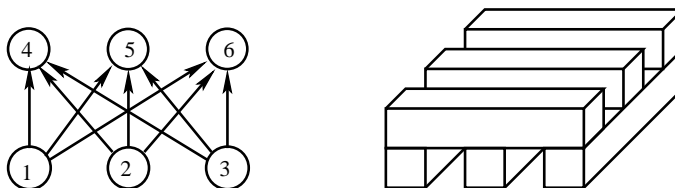


Figure 7: A non-planar graph and its representation as a three-dimensional building. A two-dimensional representation is impossible.

3.2 Stability

What about the top-left construction of Fig. 8? *Logically*, everything is OK. But it is quite probable that children won’t agree to consider it as an admissible construction since it is obviously unstable: being made of real wooden blocks it will collapse. But then, should we or should we not accept the right-hand figure? As it is, it is stable, but it is impossible to construct it in such a way that after each day’s work the intermediate result would be stable. Maybe, a reasonable way out is to put first an additional leg (like block X in bottom-left figure), and then, after putting block 3 on the top of 2, take X away.

Note that a fancy-shaped block may provide a support for two blocks constructed by two teams in parallel. In our opinion, the creativity of the students

in inventing various possibilities must be encouraged, but the teacher must always insist on formalising the rules of the game. The stability question also provides a good opportunity to discuss the notion of the center of mass.

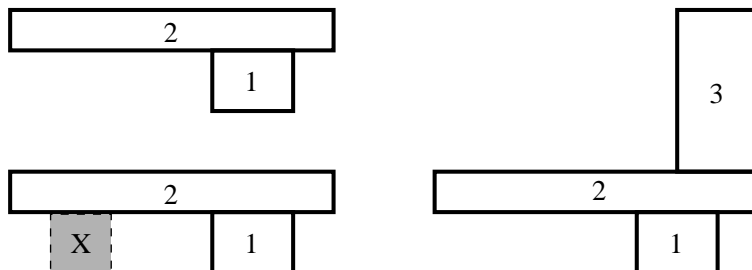


Figure 8: Making the figure stable by putting an additional support.

By the way, the center of mass M of an L-shaped figure, like in Fig. 9, may be found in a purely geometric way, without any computation: it must belong to the segment AB , and also to the segment CD ; therefore, it lies on their intersection.

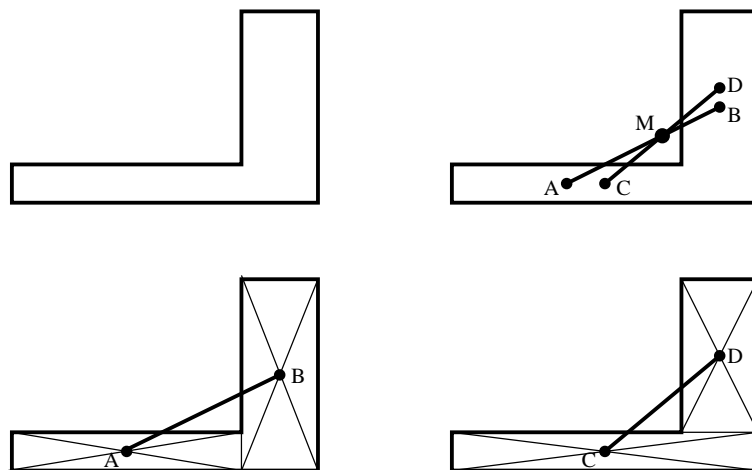


Figure 9: A geometric way of finding the center of mass of an L-shaped figure.

Verification of stability of a planar construction made of rectangular shapes may give rise to a research project for university students specialising in computational geometry. This problem is far from being simple. One must verify the stability not only of a construction as a whole but also that of its sub-constructions, though not of all of them. The question is, how to choose those which must be treated by the algorithm.

3.3 Enumeration

A *partition* of a positive integer n is a sequence $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$ of positive integers such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k > 0$ and $\lambda_1 + \lambda_2 + \dots + \lambda_k = n$. A *Young diagram* is a graphical way of representing partitions: for example, in Fig. 10, left, we see the Young diagram corresponding to the partition $\lambda = (6, 3, 3, 2, 1)$ of number $n = 15$; the numbers 6, 3, etc. are the lengths of the rows of the diagram.

A *standard Young tableau* is a Young diagram filled by the numbers from 1 to n in such a way that the sequences in every row and in every column are increasing. Fig. 10, right, provides an example of a Young tableau corresponding to the Young diagram on the left. In the notion of a standard Young tableau we easily recognize a “construction plan for one team”, with an additional but rather natural condition: the blocks in a row are constructed from left to right. The question is, how many construction plans are there for a given diagram?

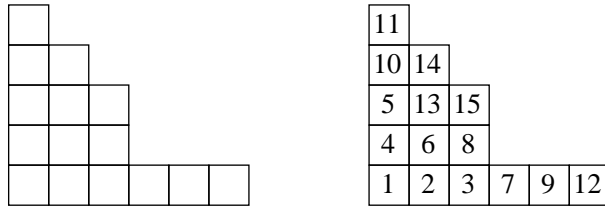


Figure 10: Left: a Young diagram; right: one of the corresponding Young tableaux.

This is not an idle question. Partitions of n , i.e., Young diagrams with n boxes, are in a bijection with the non-isomorphic irreducible representations of the symmetric group S_n . And the dimension of the representation for a given Young diagram is equal to the number of the corresponding Young tableaux.

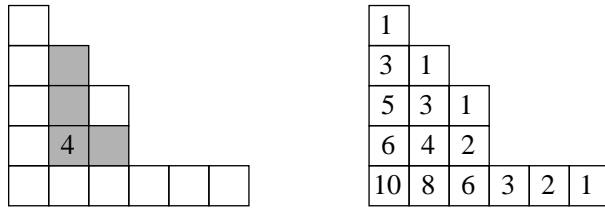


Figure 11: Left: a hook of length 4; right: all the hook lengths.

The answer is given by the famous *hook formula*. In the left-hand picture of Fig. 11, a hook is shown. Its length, that is, the number of boxes in it, is 4. The numbers which are written in the boxes of the right-hand picture of Fig. 11, show the lengths of the hooks with the corner in the corresponding box. The hook formula affirms that the number of the standard Young tableaux for

a given Young diagram is equal to $n!$ divided by the product of all the hook lengths. Thus, for the diagram of Figures 10 and 11 this number is equal to

$$\frac{15!}{10 \cdot 8 \cdot 6 \cdot 3 \cdot 2 \cdot 1 \cdot 6 \cdot 4 \cdot 2 \cdot 5 \cdot 3 \cdot 1 \cdot 3 \cdot 1 \cdot 1} = 210210.$$

All this material may be found in the book [2].

Of course, all that is far beyond the scope of matters accessible to the school-children. What is accessible, however, is experimenting with simple forms or with smaller diagrams. (Here, the knowledge of the hook formula will permit the teacher to control the results obtained by the children.) For example, for a hook-like shapes, like in Fig. 12, left, one gets the binomial coefficients. Indeed, if we are to make a standard Young tableau of such a shape having $n + 1$ boxes, namely, a corner box, k boxes above the corner, and $n - k$ boxes to the right of it, then we must put 1 to the corner, then choose k numbers to put in the column while the remaining $n - k$ numbers fill in the row.

For the diagrams made of two rows having n boxes each, like in Fig. 12, right, one gets the famous Catalan numbers, ubiquitous in enumerative combinatorics. The n th Catalan number is equal to

$$C_n = \frac{1}{n+1} \binom{2n}{n}.$$

Also, they satisfy the following recurrence:

$$C_0 = C_1 = 1, \quad C_{n+1} = C_0C_n + C_1C_{n-1} + C_2C_{n-2} + \dots + C_nC_0.$$

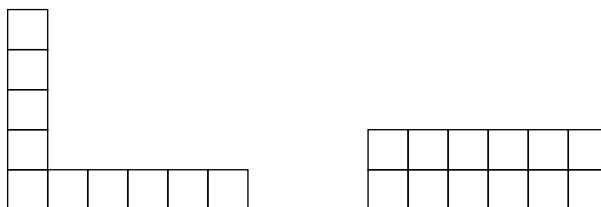


Figure 12: Left: the number of the standard Young tableaux is a binomial coefficient. Right: the number of the standard Young tableaux is a Catalan number.

In fact, the numbers themselves are not very important. What *is* important is the ability to make the enumeration not in a helter-skelter way but in an orderly manner which would ensure that nothing was lost.

* * *

We would like to finish the paper with the following suggestion to the teacher:

IMPROVISE!

References

- [1] **D. Jungnickel.** Graphs, Networks and Algorithms, Springer-Verlag, 1999.
- [2] **R. P. Stanley.** Enumerative Combinatorics, vol. **2**, Cambridge University Press, 1999.
- [3] **A. K. Zvonkin.** Mathematics for Little Ones: A Home Circle for Preschool Children, Moscow, MCCME publishers, 2007 (2nd edition; in Russian).
- [4] **A. K. Zvonkin, S. K. Lando, A. L. Semenov.** Informatica. Algorithmica: Manual for the 6th form of a junior high school, Moscow, Prosveshchenie editors, 2006 (4th edition; in Russian¹).

¹The authors are given in the Russian alphabetic order.