An algorithm:

1. Initialization - Create an initial population. This population is usually randomly generated and can be any desired size, from only a few individuals to thousands.
2. Evaluation - Each member of the population is then evaluated and we calculate a 'fitness' for that individual. The fitness value is calculated by how well it fits with our desired requirements. These requirements could be simple, 'faster algorithms are better', or more complex, 'stronger materials are better but they shouldn't be too heavy'.
3. Selection - We want to be constantly improving our populations overall fitness. Selection helps us to do this by discarding the bad designs and only keeping the best individuals in the population. There are a few different selection methods but the basic idea is the same, make it more likely that fitter individuals will be selected for our next generation.
4. Crossover - During crossover we create new individuals by combining aspects of our selected individuals. We can think of this as mimicking how sex works in nature. The hope is that by combining certain traits from two or more individuals we will create an even 'fitter' offspring which will inherit the best traits from each of it's parents.
5. Mutation - We need to add a little bit randomness into our populations' genetics otherwise every combination of solutions we can create would be in our initial population. Mutation typically works by making very small changes at random to an individuals genome.
6. And repeat. Now we have our next generation we can start again from step two until we reach a termination condition.

PseudoCode:

```
GA(S)

parametr(s): S - set of blocks
output: superstring of set S
Inicialization:
$t \leftarrow 0$
Initialize $P_t$ to random individuals from $S^*$
EVALUATE-FITNESS-GA($S$, $P_t$)
while termination condition not met
do:
        $Select\ individuals\ from\ P_t (fitness\ proportionate)$
        $Recombine\ individuals$
        $Mutate\ individuals$
        EVALUATE-FITNESS-GA($S$, $modified\ individuals$ )
        $P_{t+1} \leftarrow newly\ created\ individuals$
        $t \leftarrow t + 1$
return $(superstring\ derived\ from\ best\ individual\ in\ P_t)$
procedure EVALUATE-FITNESS-GA(S, P)
        $S - set\ of\ blocks$
        $P - population\ of\ individuals$
        for each $individual\ i \in P$
                do:
                        $generate\ derived\ string\ s(i)$
                        $m \leftarrow all\ blocks\ from\ S\ that\ are\ not\ covered\ by\ s(i)$
                        $s'(i) \leftarrow concatenation\ of\ s(i)\ and\ m$
                        $fitness(i) \leftarrow \frac{1}{\|s'(i)\|^2}$
```