Building n In A Row

Jeff Edmonds

I invented this problem to give my third year algorithms students. The original problem was to prove that the stated task is impossible. But in trying to prove it was impossible, I found a surprising algorithm for solving it. I think it fun. I hope you do too.

Formulation of Problem:
The game board is an infinitely long line of squares. In a normal round, you will place a white and then your opponent a black. The goal is to build long contiguous blocks of your colour. It is not too hard to prove that there is no algorithm that allows you to build a block longer than two. We will now develop an algorithm that builds arbitrarily long contiguous white blocks. To help, you will be given the advantage that every ten round you get to place a white but your opponent must skip his turn. (The regular expression is $((WB)^9W)^*$.)

It seemed at first to me that even if you got 100 times as many turns, that you could not build a contiguous block longer than 200, i.e. you put a 100, he caps one end by placing a black there, you place a second 100 on the other end, and he caps that end with a black. If you did manage to build a really long block, it would take you a while. Then in just two moves he can destroy it by capping both ends. Surprisingly, you can build arbitrarily long lines.

Hint given to students:
To make the algorithm easier, lets assume that the opponent is not trying to build a block of blacks. His only goal is to prevent you from building a long contiguous block of white by capping the ends of any block you are working on with a black. All blocks that we start will be far enough away from each other that they will never grow long enough to interact with each other. We will be cautious. If the opponent caps one of the ends of one of our blocks or even puts a black anywhere near that block, then we will not wait for him to cap the second end but will abandon that block completely. Hence, all we are concerned about is how many blocks we have that have no blacks near them and how long are these blocks are. The adversary each of his turns, being as greedy as he can be, will simply cap the first end of our currently longest such block. Given this, we will keep our blocks as close to being the same size as possible.

Input: The input to this problem is an integer n (say a billion).

Precondition: The board is empty.

Postcondition: You have produced a continuous block of whites of length n.

Iterations: This will be an iterative algorithm. You will have "iterations" for $i = 0, 1, \ldots, n$. Each such iteration will contain as many rounds of turns as you need to make progress and maintain the loop invariant.

Loop Invariant: After i "iterations", you have constructed many *special* blocks.

Length: Each such special block consists of i whites in a row.

Isolation: Each is far away from any black and from each other.

Number: The number of such special blocks in 10^{n-i}.

In addition to these special blocks the board will contain many white blocks that are abandoned because they have a black on one end.

- Complete Jeff's steps in completing the description/proof of this algorithm.
- Also compute the total number N of whites that you place.

Answer:

Establishing the Loop Invariant: $\langle pre\!-\!cond \rangle$ & $code_{pre\text{-}loop} \;\Rightarrow\; \langle loop\!-\!invariant \rangle$

The precondition gives that the board is empty and perhaps the value n. To establish the loop invariant for $i = 0$, we must produce 10^n blocks of whites of length zero far away from each other and far away from any black. This requires placing $N_0 = 0$ whites. It only requires deciding where the blocks will be.

Maintaining the Loop Invariant: $\langle loop\!-\!invariant' \rangle$ & *not* $\langle exit\!-\!cond \rangle$ & $code_{loop} \Rightarrow \langle loop\!-\!invariant'' \rangle$.

Suppose the LI is true for $i-1$ "iterations", i.e. the board contains 10^{n-i+1} blocks of whites each of length $i-1$ far away from each other and far away from any black. Our goal is to make these blocks one longer. For each of our next 10^{n-i+1} turns, we simply place a white next to one of the blocks of these length $i-1$ making it a block of length i. Our opponent gets nine moves for each of our ten. Hence, during these 10^{n-i+1} turns of ours he gets $0.9 \times 10^{n-i+1}$ turns. During each of these moves he can, if he wants, "destroy" one of our blocks of length i by placing a black near it. But whatever he does, he cannot destroy more than $0.9 \times 10^{n-i+1}$ of them. As required by the loop invariant, this leaves $10^{n-i+1} - 0.9 \times 10^{n-i+1} = 10^{n-i}$ blocks of length i. During this i^{th} iteration we placed $N_i = 10^{n-i+1}$ white stones.

Obtaining the Postcondition: $\langle loop\!-\!invariant \rangle$ & $\langle exit\!-\!cond \rangle$ & $code_{post\text{-}loop} \Rightarrow \langle post\!-\!cond \rangle$:

The exit condition is that $i = n$. By the loop invariant there is $10^{n-n} = 1$ contiguous white block of length n.

Measure of Progress: Clearly the measure of progress is the value i, it increases each iteration and after n such iteration the exit condition is obtained.

Running Time N: The total number of whites that you place is $N = \sum_{i=1}^{n} N_i = \sum_{i=1}^{n} 10^{n-i+1} \leq 2 \cdot 10^n$.