Opracowanie: h/
Problem H: The Great Wall Game

HISTORIA:
• wersja 1.0: 2007, Tomasz Weksej, Bogdan Yakovenko

dokument systemu SINOL 1.3.1

1 Solution

First of all, let's consider all positions where stones can be lined up

There are tree different types of them:

- All stones are lined up horisontally (there are n such different positions, one for each line)
- All stones are lined up vertically (there are n such different position, one for each column)
- All stones are lined up diagonally (there are two such position)

To find an optimal solution for each of cases above, more generalized problem can be defined.

`Problem GEN-ASSIGN`

Let $s_1, s_2, \ldots, s_n$ are different stones with coordinates $s_i = (s_x^i; s_y^i)$. In addition, $p_1, p_2, \ldots, p_n$ is a set of different points on the grid that have coordinates $p_i = (p_x^i; p_y^i)$. The problem is to move the stones from their original positions to the positions $p_1, p_2, \ldots, p_n$, so that the total number of moves is minimized. Movement's rules are the same as in the original problem.

To solve this problem, let's consider the optimal solution where stone s_1 goes to p_{k_1} position, s_2 goes to p_{k_2}, ..., s_i goes to p_{k_i}, etc. There are different ways (not always optimal) to move the stones to needed positions. If every cell were able to contain an unlimited number of stones at the same time, the optimal number of moves would be just the sum of shortest distances between s_i and p_{k_i}, i.e. $\sum_{i=1}^{n} |s_x^i - p_x^{k_i}| + |s_y^i - p_y^{k_i}|$.

At the first look, the real situation is much worse – it's not always possible to move stone by the shortest path, because other stones could block some cruicial cells. As the matter of fact, it's not a problem.

Let's consider optimal assignment and supose that stone s at position $a = (s_x; s_y)$ should be moved to position $a' = (p_x; p_y)$. One of the shortest path is $a = a_0, a_1, \ldots, a_c = a'$. Some of poits a_i could have stones on themselves. Let these points are $a = a_{b_1}, a_{b_2}, \ldots, a_{b_m}$ and $b_1 < b_2 < \ldots < b_m$.

Firstly, let's consider the situation when there is no stone at a'. To move stone from a to a' we can do the following: move a_{b_m} to a', move $a_{b_{m-1}}$ to $a_{b_m}, \cdots$, move a to a_{b_2}. The overall number of operations here is c – that is exactly the shortest path's length.

Finally, let's consider the situation when there is a stone at a'. This stone also should be moved to some position a'' . If there is no stone at position a'', the stone can be optimaly moved from a' to a'' and after this another stone from a to a'. If there is a stone at a'', let's consider sequence $h = \{a''', a^{(4)}, \ldots\}$ (where a''' is a possition when a'' shoud go, and so on) until empty field is reached and similar operation could be performed. If it's impossible to reach an empty field in h then there is cylce in one which doing nothing, but making useless moves. This situation is impossible because assignment is optimal.

Now, the problem can be easily reduced to the Assigment Problem, i.e. every $s_1, s_2, \ldots, s_n$ should be assigned to unique $p_1, p_2, \ldots, p_n$, so that sum of distances between assigned fields is minimal. There are lots of standard algorithms that can efficient solve this problem (for example Hungarian Algorithm).

Now, when the solution of GEN-ASSI problem is clear, the main problem can be solved by trying every possible allignment and calculate the cost of moves by considering every case as a sample of GEN-ASSIGN problem. As the result, $2n + 2$ GEN-ASSIGN problems needed to be solved. After a closer look at the problem, horizontal and vertical allignments are much simpler. To make an allingment vertical and horizontal moves are required. In case of vertical (horizontal) allignment, the number of vertical (horizontal) moves is constant and not depends on assignment. Taking this into account, optimal line (column) should be found first and then GEN-ASSIGN problem can be apply for one. This aproach allow us to reduce the number of GEN-ASSIGN problem runs to 4.

Indeed, horizontal and vertical allignment problems can be solved easily by greedy assignment, but it still won't prevent us from implementation of Assignment Problem, because of diagonal allignment, which can't be solved greedy. Unfortunately, this aproach has the same complexity as previous one.

The complexity of algorithm is $O(n^3)$, in case of Hungarian algorithm implementation for assignment problem. For the given parameters' restrictions this is more than enough.