

エクサウィザーズ 2019 解説

writer: semiexp, camypaper

2019 年 3 月 30 日

For International Readers: English editorial starts on page 8.

A: Regular Triangle

全ての辺の長さが等しいかどうかを判定できればよいです。例えば if 文などを用いることで、この判定を行うことができます。

C++ での解答例: <https://atcoder.jp/contests/exawizards2019/submissions/4753341>

B: Red or Blue

赤い帽子を被っている人の人数と、青い帽子を被っている人の人数を比較できれば、この問題を解くことができます。そのためには s 中に含まれる R の個数と B の個数を調べる必要があります。例えば for 文などを用いることで、これらを取得することができます。

C++ での解答例: <https://atcoder.jp/contests/exawizards2019/submissions/4753349>

C: Snuke the Wizard

説明のため、マス 0、マス $N + 1$ を定義します。これらのマス上にもはじめ、ゴーレムがいるとします。ただし、これらのマス上にいるゴーレムはどの呪文でも移動することはないとします。

各ゴーレムが最終的にどのマスに移動するかは一体あたり $O(Q)$ でシミュレーションできますが、全てのゴーレムたちの移動を愚直に実装しては当然間に合いません。各ゴーレムが最終的にどのマスにいるか、ではなく何体のゴーレムがマス 0、あるいはマス $N + 1$ にたどり着かないか、だけが重要なことを思い出しましょう。

まず、呪文によって何が起るかを観察してみましょう。すると、あるゴーレムが別のゴーレムを飛び越えることがない、即ちゴーレムたちの相対的な位置関係は変化しないことが分かります。ここから、あるゴーレムが最終的にマス 0 にたどり着くならば、そのゴーレムより左にいたゴーレムは全てマス 0 にたどり着くことが分かります。同様に、あるゴーレムが最終的にマス $N + 1$ にたどり着いたならば、そのゴーレムより右にいたゴーレムは全てマス $N + 1$ にたどり着くことが分かります。

よって、マス 0 にたどり着くゴーレムのうち、初期位置が最も右のものとマス $N + 1$ にたどり着くゴーレムのうち、初期位置が最も左のものが分かればよいです。二分法を用いることでこれらは調べることができ、 $O(Q \log N)$ でこの問題を解くことができます。

D: Modulo Operations

ある 2 つの正の整数 n, M について n を M で割ったあまりを n' とします。重要な事実として $n' \leq n$ が成立します。さらに、 $n < M$ が成立するとき限り等号が成立します。

ここから、黒板に書かれている数は、増加することはないことが分かります。また、現在黒板に書かれている数より大きい数が黒板に書かれている数を変化させることがない、ということも分かります。

以上の事実から S 中の要素のうち、最大のものが黒板に書かれた数を変化させる場合は、その要素がはじめに取り除かれた場合に限られます。よって、問題は以下のように言い換えられます。

長さ N の数列 a と整数 X が与えられる。 a は狭義単調減少であることが保証される。すぬけ君は操作を N 回行う。 i 回目の操作では確率 $\frac{1}{N+1-i}$ で X を $X \bmod a_i$ に変化させる。最終的な X の値の期待値に $N!$ をかけた値を求めよ。

問題はとてもシンプルになりました。 $f(i, x) =$ 今まで i 回操作を行い、現在の X の値が x であるときの最終的な X の値の期待値、とします。これを動的計画法を用いて求めればよいです。計算量は $O(NX)$ で、十分高速に動作します。

E: Black or White

すぬけ君が今までに黒いチョコレートを i 個、白いチョコレートを j 個食べた状態を (i, j) と書くことにします。すると、この問題は以下のように言い換えられます。

二次元平面上の $(0, 0)$ の原点にすぬけ君がいる。すぬけ君は $(+1, 0)$ という移動か $(0, +1)$ という移動を繰り返して (B, W) へ移動する。すぬけ君はどちらの移動を選んでも (B, W) へ到着できるならば、等確率で選ぶ。そうでなければ (B, W) へ到着できるような移動のみを選んで行う。

1 以上 $B + W$ 以下の各整数 i について、 i 回目の移動で $(+1, 0)$ を選ぶ確率を求めよ。

すぬけ君が確率 $\frac{1}{2}$ で移動を行わない場合は (B, j) にいる場合か (i, W) にいる場合に限られます。 i 回移動したときに $(B, i - B)$ にいる確率を p_i 、 $(i - W, W)$ にいる確率を q_i とすると、0 以上、 $B + W - 1$ 未満の各整数 i について $\frac{1 - p_i + q_i}{2}$ を計算すればいいことが分かります。

p_i, q_i は以下の漸化式で計算できます。

$$p_i = \begin{cases} 0 & (i = 0) \\ p_{i-1} + \binom{i-1}{B-1} 2^{-i} & (\text{otherwise}) \end{cases}$$
$$q_i = \begin{cases} 0 & (i = 0) \\ q_{i-1} + \binom{i-1}{W-1} 2^{-i} & (\text{otherwise}) \end{cases}$$

二項係数は前計算 $O(B + W)$ かけて計算しておけば $O(1)$ で取得できるため、この問題は $O(B + W)$ で解けました。実行時間にも十分余裕があります。

F: More Realistic Manhattan Distance

出発地点から北向きに進むと仮定したとき、目的地の位置は次の 3 通りが考えられます：

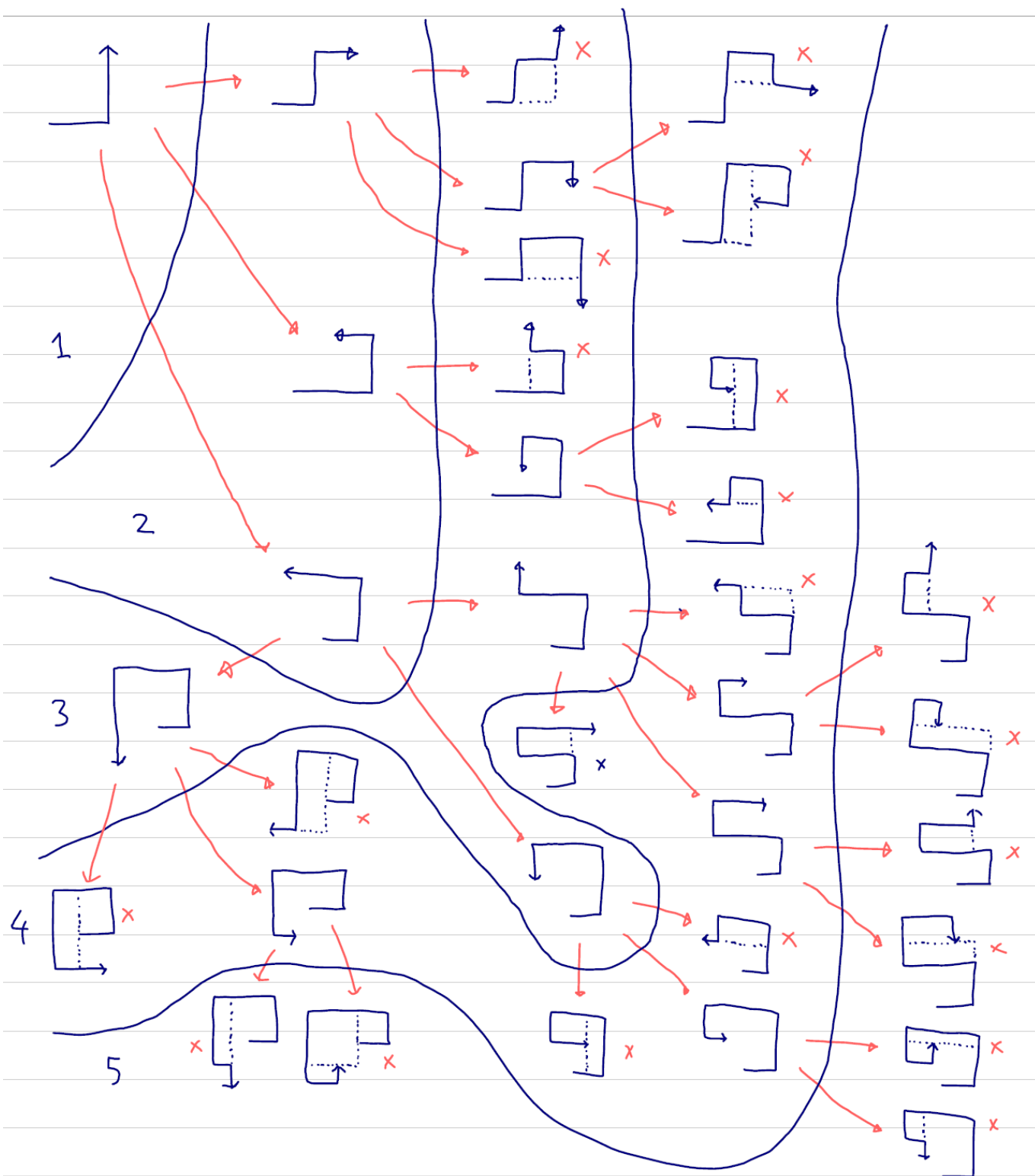
1. 出発地点から北向きにまっすぐ進んだ先にある。
2. 出発地点から北向きにまっすぐ進んだ先にはないが、出発地点よりは北にある。
3. 出発地点の真西、真東にあるか、出発地点より南にある。

(1) の場合は単にまっすぐ進むのが最適です。

(2) の場合は、目的地が出発地点の東側にあると仮定します。(西側の場合もまったく同様です)すると、次に西に曲がるとすると、西向きに通行できる最も手前の道路と交わる交差点で曲がるのが最適です。なぜならば、他の道路で西に曲がった場合、目的地までに「西向きに通行できる最も手前の道路」あるいは「出発地点から北向きに向かう道路」のいずれか片方と交わる必要がありますが、いずれの場合も他の道路で西に曲がるより距離が短い移動方法が存在するためです。また、次に東に曲がるとすると、「目的地より北にある東向きに通行できる道路で、最も南のもの」「目的地より南にある東向きに通行できる道路で、最も北のもの」のいずれかで曲がるのが最適であることが、同様にわかります。

(3) の場合は、次に西に曲がるとすると、西向きに通行できる最も手前の道路と交わる交差点で曲がるのが最適です。東向きについても同様です。なぜならば、出発地点から北向きに進んでから目的地に至るような任意の経路は、西向きないし東向きの最も手前の道路と交わるためです。

よって、曲がる回数が K 回以下であると仮定すると、 $O(3^K)$ の経路を調べれば十分であることがわかります(最も手前の道路などは、適切に前計算をしておくとも $O(1)$ で計算可能です)。また、曲がる回数は 4 回以下として良いということが示せます。これは、経路の折れ曲がり方を、短絡可能だったり(経路長を変えずに)折れ曲がる回数を減らせたりする経路を除外していきながらすべて調べることで確かめられます。詳しくは次の図を見てください。(点線は、可能な短絡路を表します。道路の通行可能な向きによって異なる方法で短絡が発生する場合も含まれています。)



ExaWizards 2019 Editorial

writer: semiexp, camypaper

March 30, 2019

A: Regular Triangle

The problem can be solved by checking if all the sides have equal lengths. This can be done by, for example, an “if” statement.

Sample solution in C++: <https://atcoder.jp/contests/exawizards2019/submissions/4753341>

B: Red or Blue

The problem can be solved by comparing the number of people with red hats and the number of people with blue hats. To do that, we first need to find the number of R and B in s . This can be found by, for example, a “for” statement.

Sample solution in C++: <https://atcoder.jp/contests/exawizards2019/submissions/4753349>

C: Snuke the Wizard

For convenience, let us define Square 0 and Square $N + 1$, and assume that there is a golem also on these squares initially, but no spells move the golems on these squares.

We can simulate the moves of each golem and find its final position in $O(Q)$ time, but there is obviously not enough time to do this for all the golems. Remember that we are only interested in how many golems end up somewhere except Square 0 and Square $N + 1$, not the final position of each golem.

Let us observe what happens in a spell. First, we can see that a golem never jumps over another, that is, the relative order of the golems does not change. Thus, if a golem ends up in Square 0, all golems to the left of that golem end up in Square 0. Similarly, if a golem ends up in Square $N + 1$, all golems to the right of that golem end up in Square $N + 1$.

Therefore, we can solve the problem by finding the golem ending up in Square 0 with the rightmost initial position, and the golem ending up in Square $N + 1$ with the leftmost initial position. These can be found by binary search, which solves the problem in $O(Q \log N)$ time.

D: Modulo Operations

For two positive integers n and M , let us denote $X \bmod a_i$ by n' . One important fact is that $n' \leq n$. Additionally, the equality holds if and only if $n < M$.

Thus, it can be seen that the number on the blackboard never increases, and that it is never changed by a number greater than its current value.

From these facts, the largest element in S changes the number on the blackboard only if that element was removed first, and the problem can be rephrased as the following:

Given is a sequence a of length N and an integer X . It is guaranteed that a is strictly monotonically decreasing. Snuke will perform N operations. In the i -th operation, he will change X to $X \bmod a_i$ with probability $\frac{1}{N+1-i}$. Find the expected final value of X , multiplied by $N!$.

Now the problem is very simple. Let $f(i, x) =$ (the expected final value of X when i operations are performed so far and the current value of X is x). The problem can be solved by finding these values by dynamic programming. The time complexity is $O(NX)$, which is small enough.

E: Black or White

Let (i, j) denote the state where Snuke has eaten i black pieces and j white pieces so far. Then, the problem can be rephrased as the following:

Snuke is at the origin $(0, 0)$ in the two-dimensional plane. He will go to (B, W) by repeatedly performing one of the two moves: $(+1, 0)$ and $(0, +1)$. When both moves can lead to (B, W) , he chooses one of them with equal probability. Otherwise, he will only choose the move that leads to (B, W) .

For each integer i from 1 to $B + W$, find the probability that $(+1, 0)$ is chosen in the i -th move.

He will choose each move with probability $\frac{1}{2}$, with the only exception of the case in which he is at (B, j) or (i, W) . Let p_i and q_i be the probability that he is at $(B, i - B)$ and $(i - W, W)$ after i moves, respectively. Then, it can be seen that the answer for each i is $\frac{1 - p_i + q_i}{2}$.

p_i and q_i can be computed from the following recurrence relation:

$$p_i = \begin{cases} 0 & (i = 0) \\ p_{i-1} + \binom{i-1}{B-1} 2^{-i} & (\text{otherwise}) \end{cases}$$
$$q_i = \begin{cases} 0 & (i = 0) \\ q_{i-1} + \binom{i-1}{W-1} 2^{-i} & (\text{otherwise}) \end{cases}$$

Since a binomial coefficient can be obtained in $O(1)$ time with proper pre-computation with $O(B + W)$ time, the problem is now solved in $O(B + W)$ time, which is fast enough.

F: More Realistic Manhattan Distance

Assuming that we go north from the place of departure, the destination is at one of the following three locations:

1. Can be reached by going straight north from the place of departure.
2. Cannot be reached by going straight north from the place of departure, but to the north of it.
3. Straight west or east from the place of departure, or to the south of it.

In case (1), it is optimal to just go straight.

In case (2), let us assume that the destination is to the east of the place of departure. (The argument will be the same for the west case.) Then, if the next turn we take is to the west, it is optimal to take that turn at the intersection of the current road and the first westward road we will reach. This is because, if we take the turn at a different intersection, we need to cross either “the first westward road” or “the current road” before we reach the destination, but in both cases there exists a way to reach there with a shorter distance than taking the turn at that different intersection. Similarly, if the next turn we take is to the east, it is optimal to take that turn either at the intersection of the current road and the southmost eastward road that is to the north of the destination, or at the intersection of the current road and the northmost eastward road that is to the south of the destination.

In case (3), if the next turn we take is to the west, it is optimal to take that turn at the intersection of the current road and the first westward road we will reach, and this is the same if the next turn we take is to the east. This is because, any path from the place of the departure to the destination that first goes north, crosses “the first westward (or eastward) road”.

Therefore, assuming that we take at most K turns, we only need to check $O(3^K)$ paths (we can compute “the first roads” and such in $O(1)$ time by proper pre-computation). Additionally, we can also show that it can be assumed that we take at most 4 turns. This can be seen by enumerating all possible ways in which a path take turns, omitting paths that can be shortened or can be made to have fewer turns (without affecting the length). See the following figure for detail. (Dotted lines represent the possible shortened paths. There are cases where there will be different shortened paths depending on the direction of the roads.)

