ICPC 2025 国内予選 問題解説

Problem A: 2025

原案: 稲葉 一浩　　主担当: 鵜川 始陽

式 $\sum_{a=1}^{n}\sum_{b=1}^{n} ab$ を計算すればよい．一般式を求めると $\frac{1}{4}n^2(1+n)^2$ になるが，この問題は n が小さいのでそこまでする必要はない．素朴に二重ループで全ての積 ab を求めても十分に間に合う．

Problem B: 接頭辞と接尾辞を同じに

原案: 楠本 充　　主担当: 城下 慎也

文字列 s を 2 個用意し，それらをそのまま連結した文字列 ss は明らかに s を接頭辞と接尾辞の両方に含む．よって最小の文字列の長さは $2n$ 以下である．

長さ $n+k$ $(1 \leq k \leq n-1)$ の文字列 t_k が条件を満たすとする．このとき，t の先頭から n 文字 $(=s)$ と末尾から n 文字 $(=s)$ のうち，t_k の中心側に近い端点から $n-k$ 文字同士が重複するため，s の先頭から $n-k$ 文字と末尾から $n-k$ 文字は一致しなければならない．逆にそのような s に対しては必ず t_k を一意に構成できる．k を 1 から順に試し，最初に見つかった k に対応する t_k が正解となる．そのような k がない場合は ss が正解となる．

Problem C: 働き者のカレンダー

原案: 平原 秀一　　主担当: 平原 秀一

今日（月曜日）から始まる m 日間のうち，働ける日数を求める問題である．土日は休日で，さらに追加の休日が与えられる．

まず，m 日間に含まれる平日（月〜金）の総数を計算する．週の周期性から，平日数は $\lfloor m/7 \rfloor \times 5 + \min(m \bmod 7, 5)$ となる．次に，追加の休日のうち平日に当たるものを数える．d 日目が平日であるための条件は $1 \leq (d \bmod 7) \leq 5$ である．平日の総数から平日の休日数を引けば答えが得られる．

この問題の主な注意点は，m と a_i が最大 10^{18} という大きな値を取ることである．C++では `long long` 型を使用し，Python では整数除算 `//` を使用する必要がある．また，追加の休日は重複する可能性があるため，重複を除去する必要がある．

Problem D: 古代のゲーム盤

原案: 隈部 壮　　主担当: 住谷 達哉

遺物が学説と矛盾しないための必要条件として，各領域が正方形状であることと同じ大きさであること以外は，この残存する盤面自体も問題文の条件を満たす必要がある．すなわち，同色の長方形領域（同じ大きさでなくてもよい）がチェス盤のように前後にも左右にも白黒交互に並ぶ配色になっている必要がある．

これは次のように判定できる．まず，各行に関して，同色のタイルが連続する数を左から順に並べた数列を作る．例えば ‘..###...#’ ならば $(2, 3, 3, 1)$ が得られる．このようにして各行から得られる数列は全て一致する必要がある．各列に関しても，同様に作られた数列が全て一致する必要がある．この二つが成り立つことと上述の条件が満たされることは同値であり，以下その場合を考える（そうでない場合は -1 を出力する）．

各領域が正方形状であることと同じ大きさであるという条件を戻して，そのような盤面の一部になっているかを判定する．各行から得られる数列を $(a_1, \ldots, a_n)$ とする（前提よりどの行を選んでも同じ数列が得られる）．同様に各列から得られる数列を $(b_1, \ldots, b_m)$ とする．正方形領域の一辺のタイル数を k とすると，遺物の端の部分は元の正方形領域が切り取られている可能性があることを考慮し，$k \geq a_1, a_n, b_1, b_m$ が成り立つ．端以外は切り取られていないため，$k = a_i\ (2 \leq i \leq n-1)$, $k = b_j\ (2 \leq j \leq m-1)$ が成り立つ．これらを満たす k が存在しないならば -1 を，存在するが一意ではないならば 0 を，一意に存在するならばその値を出力すればよい．

Problem E: 航路廃止

原案: 稲葉 一浩　　主担当: 稲葉 一浩

入力を星 1 を根とする木として見たとき，ある星に辿りつく航路の期限が e だとすると，その親に辿りつく航路には実質的に期限 $e-1$ が課されていると考えることができる．この考えに基づき，子から親へ，元々入力で与えられた期限との小さい方を取りながら伝搬させていくと，親方向に向かうほど早い期限が設定された木が得られる．

あとは単純に，この木で期限の早い航路から順に使っていけば，解が（存在するならば必ず）得られる．この貪欲法である航路の期限 e を過ぎてしまうということは，期限 e 以下の辺が $e+2$ 本以上あるということで，他のどのような順で並べようとも，すべての期限に間に合う方法はない．

Problem F: 犬の芸

原案: 隈部 壮　　主担当: 隈部 壮

‘a’ と ‘b’ のみからなる文字列 S に特定の操作を繰り返して，文字列 T にできるかを問う問題である．操作は以下の二種類である．

- 先頭から見て初めて現れる ‘a’ を ‘b’ に書き換え，その後初めて現れる ‘b’ を ‘a’ に書き換える．
- 先頭から見て初めて現れる ‘b’ を ‘a’ に書き換え，その後初めて現れる ‘a’ を ‘b’ に書き換える．

各操作は文字列中に現れる ‘a’ の個数を変えないので，S と T に現れる ‘a’ の数が異なる場合，S を T に一致させるのは不可能である．以下，S と T に現れる ‘a’ の数は（したがって ‘b’ の数も）等しいとする．この場合，以下の操作列によって S を T に変換できることが証明できる．

まず，二つ目の操作を可能な限り繰り返すことで，S を aa...aabb...bb の形にできる．その後，末尾から順に S を T に揃えていく．具体的には，S と T の i 文字目が異なるような最大のインデックス i に対し，

- S の i 文字目が ‘a’ であれば，二つ目の操作を S の i 文字目が ‘b’ になるまで繰り返す
- S の i 文字目が ‘b’ であれば，一つ目の操作を S の i 文字目が ‘a’ になるまで繰り返す

という操作を行うことで，$i+1$ 文字目以降を保ったまま，S と T の i 文字目を揃えることができる．したがって，この操作を繰り返すことで，最終的には S を T に一致させることができる．操作列の長さは大雑把に見積もっても n^2 回以下であるため，問題の制約に合致する．

Problem G: 面の数

原案: 楠本 充　　主担当: 楠本 充

凸多面体を取ったとき，H_1 と H_2 に描いた凸多角形が上面と下面に対応し，それ以外が側面となるものができることがわかる．側面には三角形か四角形のどちらかの面ができる．

四角形の面ができる条件を考えることにする．これが起きるのは 2 つの多角形に時計回りで向きを付けたときに，向きが同じがあるときだけである．このとき，同じ向きの 2 辺を境界に持つような四角形が凸多面体の面として現れる．それ以外のときは三角形の面が構成される．この考察では描いた多角形の辺の向きだけが大事で，辺の長さや頂点の位置は重要ではない．

どの辺の向きも一致しないケースが常に存在し，その場合の面の個数は $n+m+2$ になる．そうでないときは，上面の辺と下面の辺の組で，向きが同じであるものが存在する．同じ向きになる辺の組を固定すると，残りの辺の向きが決まり，全体でいくつ向きの同じ辺ペアが存在するかわかる．これにより，ありうる面の数を列挙できる．

計算時間は $O(nm(n+m))$ などで済む．

Problem H: カッコ

原案: 岡 智洋　　主担当: 岡 智洋

ここでは開きカッコを L，閉じカッコを R と書くことにする．正しいカッコ列の長さは偶数であるので最初のスタンプの位置と最後のスタンプの位置の偶奇は異なることがわかる．まずは最初のスタンプが最後のスタンプより左にある場合を数える．

2 つのケースに分ける．(i) 最初のスタンプから最後のスタンプまで最短距離のカッコの列が L と R が交互になっている場合，これは入力データを先頭から 1 文字ずつ見ていってカウンターを更新しながら数えることができる．つまり，同じ文字が連続したときはカウンターを 0 にリセットし，L を読むときはカウンターを 1 増加する．R を読むときにカウンターの値だけ，解に加算する．

(ii) 最初のスタンプから最後のスタンプまで最短距離のカッコの列が交互になっていない場合，印字される列が正しいカッコ列になるためには，最初のスタンプの条件：最初のスタンプから 2 つ連続した L を印字するするまでに 2 つ連続した R を通過しないこと，および，最後のスタンプの条件：2 つ連続した R を印字してから最後のスタンプまでに 2 つ連続した L を通過しないことがある．2 つ連続した L と 2 つ連続した R は適切な数になるように増やすことができるので，その中間のカッコ列には条件はない．各スタンプについてこの 2 つの条件を満たしているかをそれぞれ，あらかじめ計算しておく．そうして (i) と重複しないように気をつけながら数えるには，入力データをスキャンしながら，最初のスタンプの条件を満たしたものを「LR 交互を継続しているもの」と「交互ではなくなったもの」，スタンプの位置の偶奇も気にするので 4 つの変数に分けてカウントし，最後のスタンプの条件を満たしたものを読むときに解に加算する．

最初のスタンプが最後のスタンプよりも右にある場合を数えるには，入力データを逆順にしてから同様に数えれば良い．

Problem I: 昼食会の準備

原案: 佐藤 遼太郎　　主担当: 佐藤 遼太郎

まず，同じ種類の弁当同士の交換は一切行う必要がない．よって，ある種類の弁当が $2k$ 個あったとき，最適解では各 $i = 1, \ldots, k$ について最終的にその種類の弁当の中で i 番目と $k+i$ 番目のものが向かい合うことになる．つまり，初期状態が与えられた時点で最終的にどの弁当とどの弁当が向かい合うべきかがすぐ分かる．

ここで，円周上に $2n$ 個の弁当が並んでいる状況において，最終的に向かい合うべき弁当のペア同士を結ぶ n 本の線分を引き，これらの線分のペアのうち互いに交わるものの個数を考えると，この値は以下のような性質を持つことが証明できる：

- この値は一回の操作で高々 1 しか変わらない.
- この値は目標とする最終状態で最大値 $n(n-1)/2$ をとる.
- 目標とする最終状態ではない任意の状態について，この値を 1 増やす操作が必ず存在する.

よって与えられた初期状態での互いに交わる線分のペアを数えて，$n(n-1)/2$ から引いた値が答えとなる．この数え上げは Fenwick tree などのデータ構造を適切に使えば $O(n \log n)$ でき，実装上も十分高速に動く.