審判長講評

山口 勇太郎（大阪大学）

全体コメント

まず 2 つの統計データを示す．表 1 は「正解数ごとのチーム数」であり，表 2 は「各問題の正答・誤答チーム数と合計提出数」である．なお，審判団が想定した難易度は大まかには以下の通りであり，表 2 では問題をこの順に並べている．

$$A < B < E < K \leq C < I \leq L < D < F \leq G \ll H \leq J$$

表 1: チーム数 vs. 正解数

正解数	12	11	10	9	8	7	6	5	4	3	2	1	0
チーム数	0	1	1	4	5	8	17	8	6	5	0	0	0
累積チーム数	0	1	2	6	11	19	36	44	50	55	55	55	55
チーム数（凍結前）	0	0	1	1	3	11	10	15	7	7	0	0	0
累積チーム数（凍結前）	0	0	1	2	5	16	26	41	48	55	55	55	55

表 2: 正答・誤答チーム数，合計提出数 vs. 問題（想定難易度順）

問題	A	B	E	K	C	I	L	D	F	G	H	J
正答チーム数	55	55	54	41	50	36	8	22	4	8	0	1
誤答チーム数	0	0	1	12	2	5	5	1	10	10	1	1
合計提出数	79	89	66	133	68	89	27	28	20	56	1	4
正答チーム数（凍結前）	55	55	52	40	46	31	4	15	3	3	0	0
合計提出数（凍結前）	79	89	62	97	60	71	15	13	3	12	0	0

はじめに，全チームが A, B の両問題に正解したことと，3 問以上の問題に正解したことを称えたい．

昨年からの大きな変更点として，問題数を例年の 11 から 12 に増やしたことが挙げられる．背景には，昨年の優勝チームが 1 時間近くを残して全問正解を成し遂げたという事実や，世界大会と合わせようとしたという意図もあったが，それ以上に「上から下まで全ての参加チームがなるべく時間いっぱい楽しめる（やることが無くならない）ようにしたい」という気持ちがあった．（これは国内予選についても同じである．）順位表凍結後の 1 時間における（意味のある）提出数は 146 (/660)，1 回以上提出したチーム数は 49 (/55)，1 問以上正解したチーム数は 26 (/55) であり，数字に現れず実際には分からない部分もあるが，多くのチームに最後の一瞬まで楽しんでもらえていれば幸いである．

問題セットとしては，1〜3 位で正解数が 1 問ずつ違うことや，3〜9 問正解のチーム数がかなり綺麗に分布していることを見ると，完成度の高いものであったと言えよう．難易度設定は，A, B が簡単，E, K, C が序盤，I, L, D が中盤（ Asia Pacific Championship 進出争い），F, G が終盤（入賞争い），H, J が防衛（優勝争い）という想定であった．概ね想定通りの結果となったが，中盤枠の 3 問に関しては，I, D が多く

解かれて L があまり解かれない側に偏った結果となった．この 3 問のジャンルはかなりバラけており，正解数から受ける印象ほど大きな難易度差は無く，時間制限と初動の影響を大きく受けたものであると考えられる．また，H と J は非常に難しい問題であり，手前の 10 問の歯応えもあって,「それぞれ時間内に正解が出るかどうか」「全体としては 10 問正解で優勝争いになるか」というのが開始直前時点での大方の見方であった．蓋を開けてみれば，優勝チームは H, J 以外の 10 問を順位表凍結前に通し切り，残り時間で J に正解して単独 11 問正解となるという，予想を超えた素晴らしい立ち回りであった．

問題別コメント

以下では，各問題について簡単に解説する．各問題には原案提案者と準備の主担当者を併記するが，原案のレビューや問題の選定，問題文の校正，正答・誤答プログラムの作成，入出力チェックなど，どの問題にも多くの審判が関わっており，全体として審判団全員で作り上げた問題セットであることを強調しておく．

A: Ribbon on the Christmas Present

原案: 稲葉 一浩　　主担当: 鵜川 始陽

1 本のリボンに濃淡の違う赤色を塗って指定のグラデーションを作りたいが，塗る操作の回数をなるべく少なくしたいという問題である．1 回の操作で好きな長さを塗ることができ，薄い赤に濃い赤を上塗りすることはできるが，その逆はできない．実際には，各濃さで塗る区間を決めて，薄い方から順に塗ればよい．また，今塗る濃さよりも濃い部分は後で上塗りされるので，塗りたいなら塗ってしまえばよい．塗る濃さをスタックで管理しながらリボンを 1 回走査することで，このような貪欲な塗り方は長さの線形時間で計算できる．この問題では長さも濃さも 100 以下の整数なので，計算量を頑張って削減する必要はあまり無く，濃さごとに塗るシミュレーションをするなどの素朴な方法でも十分間に合うだろう．

最終的に全てのチームが正解したが，1 問目としては想定より多くの誤答の提出があった．典型的な誤答として，上記の「後で上塗りされるので，塗りたいなら塗ってしまえばよい」を本来不要な部分にまで適用してしまい，余計な操作をしているものなどがあった．

B: The Sparsest Number in Between

原案: 柴山 悦哉　　主担当: 柴山 悦哉

a 以上 b 以下の正整数であって，2 進法表記で 1 が最も少ないもののうち最小のものを求める問題である．解の候補が最大 10^{18} 個あるので，素朴に全てをチェックする訳にはいかない．候補を絞る方法としていくつかの考え方があるが，まず a と b の上位桁で共通する部分は固定してしまってよい．すると，$a < b$ であれば，共通の上位桁部分を X として $a = X0\ldots_{(2)}$, $b = X1\ldots_{(2)}$ という形になっているはずなので，$x = X10\ldots0_{(2)}$ は $a < x \leq b$ を満たす．この x が解の候補なので，他に候補となり得るものは，上位桁が X で残りの位に 1 をたかだか 1 個持つような整数のみである．そのような整数はたかだか 60 個程度であるので，これらと a, b との大小関係を全てチェックすることでこの問題に正解できる．

この問題もそれなりに誤答の提出はあったものの，最終的には全てのチームが正解しており，出場チームのレベルの高さが伺える．

C: Omnes Viae Yokohamam Ducunt?

原案: 北川 宜稔　　主担当: 北川 宜稔

ラテン語の「全ての道はローマに通ず」を捩った問題名であり，問題文の読解の比重が大きい問題であったと思われる．要約すれば，各頂点 v に重要性 p_v，各辺 e に脆弱性 q_e がそれぞれ定められた連結無向グラフにおいて，以下の値が最小になるような全域木 T を求める問題である．

$$f(T) = \sum_{e \in T} q_e \sum_{v \in X_{T,e}} p_v$$

$$\text{ただし } X_{T,e} \coloneqq \{\, v \mid v \text{ は } T - e \text{ において頂点 } 1 \text{ と異なる連結成分に属する頂点} \,\}$$

一見複雑な目的関数に見えるが，和の順序を入れ替える（いわゆる「主客転倒」と呼ばれる考え方を使う）ことで，次の形に書き換えることができる．

$$f(T) = \sum_{v=1}^{n} p_v \sum_{e \in P_{T,v}} q_e$$

$$\text{ただし } P_{T,v} \coloneqq \{\, e \mid e \text{ は } T \text{ において頂点 } 1 \text{ と頂点 } v \text{ を結ぶパス上にある辺} \,\}$$

こうしてみると，後者の和は「各 q_e を辺長と見なしたときの T における頂点 1 から頂点 v へのパス長」であり，頂点ごとにこれを最小化できるとよい．したがって，Dijkstra 法で最短路木（各頂点への最短パスを含むような全域木）を構築することで，求めたい答えを計算できる．適切に優先度付きキューを用いて実装すれば計算量は $\mathrm{O}(m \log n)$ 時間となり，この問題に正解できる．

難易度としては序盤枠であり，実際に最終的には 50 チームが正解したものの，問題文の読解が他の問題に比べて重めなこともあり，初動は遅れ気味であった．

D: Tree Generators

原案: 楠本 充　　主担当: 楠本 充

構文解析要素を含む問題であるが，その部分は括弧列 $(+\alpha)$ から二分木を作るだけでさほど重くなく，本質はその後の考察である．ただし，サイズが 1 の入力もあり得て，その場合には入力に括弧が含まれないので，これがコーナーケースになるような実装をしてしまわないように多少の注意が必要である．

与えられた二分木について，その葉に 1〜n のラベルを左から順に割り振って，各中間ノードについて「左右の部分木に含まれる葉のラベルを任意に 1 つずつ選んで，その間に辺を張る」という操作を行うと，（操作で張られた辺に関して）n 頂点のラベル付き木が生成される．このとき，与えられた 2 つの二分木から共通に生成され得るようなラベル付き木の個数を数え上げる問題である．

重要な観察として，以下の非常に綺麗な事実が成り立つ．（証明は二分木の深さに関する帰納法でできる．）

- 二分木 T の各中間ノード v について，左部分木の右端の葉のラベルを $\ell_T(v)$ とする．このとき ℓ_T は，T の中間ノード全体と $\{1, 2, \ldots, n-1\}$ の間の 1 対 1 の対応を定める．
- 二分木 T_1, T_2 から共通に生成されるラベル付き木について，辺 e がそれぞれの中間ノード v_1, v_2 について張られるとすると，$\ell_{T_1}(v_1) = \ell_{T_2}(v_2)$ が成り立つ．

したがって，ℓ_{T_1}, ℓ_{T_2} で定まる値が同じ中間ノードの各ペアについて，それらが同じ辺を張るような場合の数を求めればよい．これは，各中間ノードの左右の部分木について，ラベル集合の共通部分のサイズの積を計算すれば求まるので，全体として入力サイズの線形時間でこの問題を解くことができる．

中盤問題の中では難しめだと想定していたが，L よりもだいぶ早く最初の正解が出たことや，上の「事実」に気付いてしまいさえすれば厳密な証明は必要無いことなどもあり，想定よりも多く解かれる結果となったと考えられる．

E: E-Circuit Is Now on Sale!

原案: 城下 慎也　主担当: 城下 慎也

2 次元グリッド上に計算木が与えられ，それを正しく認識して計算結果を求める問題である．入力サイズが 50×50 以下と小さく，計算途中の値も 10^{18} を超えない非負整数であることが保証されており，良心的な設定であろう．根から深さ優先的に再帰して計算するなり，葉から根に向かって計算結果を送っていくなり，各自のやりやすい方針で実装するのが望ましい．考察より実装というタイプの問題であり，コーナーケースなどの罠も特に無いが，ケアレスミスやオーバーフローには要注意である．

F: The Farthest Point

原案: 山口 文彦　主担当: 山口 文彦

言わずと知れた「小谷の蟻」から派生した問題であり，直方体表面における頂点からの最遠点を求める問題である．問題文の読解という意味では B や I と並んで最も簡単かもしれないが，考えてみると意外と難しいというタイプの問題であろう．代表的な嘘解法として，目的関数の単峰性を期待した局所探索や三分探索などが挙げられるが，実は単峰とは限らないので素朴な山登り法などは撃墜されてしまう．

正解に至るには，数学（算数？）的な観察を積み重ねるのがよい．まず,「最遠点は頂点自身が属する 3 つの面の外にある」ということはほぼ明らかである．重要な観察は,「最遠点では，頂点からそこへ至る複数の最短経路の長さが釣り合って等しくなる」ということである．これに気付けば,「展開図上で異なる位置に配置された同一頂点から距離が等しくなるような点が最遠点の候補である」ことが分かり,「展開図上で同一頂点の（仮想的な）配置としてあり得る三角形の外心を求めて，その中で適切な位置にあるものを選べばよい」という結論に至るであろう．

実装量も考慮して G より正解数が多くなると想定されていたが，実際には G の方が先に正解されたこともあり，勇敢に G を解きに行くチームが思いの外多かったようである．

G: Beyond the Former Explorer

原案: 森田 晃平　主担当: 森田 晃平, 楠本 充

インタラクティブ問題である．$(2n+1) \times (2n+1)$ の正方形グリッド状のフィールド上にスタートからゴールへの移動経路が 1 つ与えられており，その経路上のマスに到達すると移動方向が分かるが，ゴールに至るまでの移動回数を抑えなければならないという問題である．王道かつ典型的な戦略として，二分探索がある．フィールドを縦断（あるいは横断）するように移動経路を確認すれば，その移動回数からゴールがどちら側にあるかを認識できる．これを繰り返すことでゴールが存在し得る領域を半々に絞り込めるので，$\mathrm{O}(n \log n)$ 回の移動でゴールに辿り着ける．

しかし，この問題では $n = 2000$ の入力に対して 30000 回以下の移動でゴールに辿り着くことを要求されており，領域を半分にするために毎回 $2n$ 回ずつ移動していたのでは間に合わない．よく考えてみると，領域を半分にするために縦と横の短い方で切ることにすれば，必要な移動回数は 2 回ごとに半分になっていき，総和は $\mathrm{O}(n)$ 回となる．このアイデアに基づいて適切に実装すれば，この問題に正解できる．

簡単に書いたが，この問題の本質は実装であり，上述のアイデアを如何に簡潔にミスなく実装し切るかを問われている問題であると言える．その大変さも考慮して,「もしかしたら 1, 2 チーム程度しか正解しないのではないか（F よりも H, J 寄りになるかもしれない）」という意見もあったが，実際には 8 チームが正解し，正解できなかった中に惜しいチームもあったように見受けられる．予想を遥かに超えた健闘を称えたいと思う．

H: Remodeling the Dungeon 2

原案: 山口 勇太郎　　主担当: 佐藤 遼太郎

防衛枠の最難問の 1 つである．2 年前の問題 G に引き続き，女王様が $h \times w$ の長方形グリッド状のダンジョンを改築したいという問題である．今回の設定では，ダンジョンは連結なグリッドグラフであり，いくつかの通路を埋めて壁にする（辺を削除する）ことで全域木にしたい．ただし，結果として得られる木において，任意の葉間の距離が偶数であるようにしなければならない．

まず，グリッドは 2 部グラフであるので，結果として得られる全域木における頂点集合の 2 分割は常に固定されている．したがって,「任意の葉間の距離が偶数」という条件は,「固定された 2 分割において，葉が全て同じ側に属する」と言い換えられる．この頂点集合の 2 分割が等分であれば目的の達成は不可能である．なぜなら，葉が無い側の頂点の次数が全て 2 以上となり，辺数が頂点数以上となって木になれないからである．この観察から，一方の頂点集合が他方より真に大きく，大きい側に全ての葉が属するようにしなければならないことが分かる．以下では，小さい側の頂点集合を A，大きい側の頂点集合を B とする．

次の観察として，求めたいものは全域木であるが，これを森に緩めても等価である．なぜなら，A 内の頂点の次数が全て 2 以上であるような森が存在すれば,（元のグリッドの連結性より）それに辺を足して全域木にすることができるからである．さらに,「A 内の頂点の次数が全て 2 以上」という条件を「A 内の頂点の次数が全てちょうど 2」と強めても等価である．なぜなら，前者を満たす森から「A 内で次数が 3 以上の頂点に接続する辺」をいくつか削除すれば後者を満たす森が得られるからである．

最終的に，求めたいものは「A 内の頂点の次数が全てちょうど 2 であるような森」と言い換えられた．そのような森では，B 内の葉とその隣接頂点のペアを 1 つずつ選んで削除していくことで，A 側から見た完全マッチングを作ることができる.（これは $|A|$ に関する帰納法で証明できる.）したがって,「最大マッチングが A 内の頂点を全てカバーする」ことは，求めたい森が存在するための必要条件である．

最後に，その必要条件が満たされた上で，求めたい森がどういう形をしているかを考えよう．森内の最大マッチング M を 1 つ固定したとき，A 側から見れば，各頂点には M の辺と M 以外の辺が 1 本ずつ接続していることになる．M の辺を A から B に，M 以外の辺を B から A に向き付けると,（閉路が無いので）この条件は「A 内の任意の頂点は，マッチングに参加していない B 内のある頂点から到達可能」と言い換えられる．実はこの条件が最大マッチング M の取り方に依らず必要十分であることが証明でき（いわゆる Dulmage–Mendelsohn 分解に思いを馳せるとよい），これを入力グラフに対して確認してそのような森を構築することでこの問題に正解できる．

計算量は 2 部グラフの最大マッチングを求める部分がボトルネックであり，Hopcroft–Karp のアルゴリズム（最大流の Dinic のアルゴリズム）を用いることで $\mathrm{O}((hw)^{1.5})$ となる．なお,「片側頂点の次数が全てちょうど 2 であるような森」はマトロイド交叉の典型例であり，多項式時間で求めるだけであればそれでもよいが，$hw \approx 400^2$ のケースで制限時間 8 秒に間に合うほど高速に動作する実装は絶望的であろう．

本コンテストでは正解チームが出ず残念な気持ちもあるが，J とともに防衛枠の問題であったことと，ミラーコンテストでは H, J ともに 1 チームずつが正解したことを考えると，与えられた役割を果たしていたとも言えよう．

I: Greatest of the Greatest Common Divisors

原案: 岡 智洋　　主担当: 岡 智洋

正整数列といくつかの区間が与えられ，各区間内にある異なる 2 つの整数の最大公約数のうち最大のものを求める問題である．列の長さが n，区間の数が q，整数の最大値が $a_{\max}$ のとき，ユークリッドの互除法を用いて愚直に確認すれば $\mathrm{O}(qn^2 \log a_{\max})$ 時間で正しく答えることができるが，もちろんこれでは間に合わない．各ペアについて前計算しておけば $\mathrm{O}(n^2 \log a_{\max} + qn)$ 時間などに落とすことは難しくないが，この問題では n, q が 10^5 程度まで大きくなり得るので，これでもまだ不十分である．

重要な言い換えは,「区間内の異なる 2 つの整数の公約数は，区間内の整数の約数として複数回現れる整数である」ということである．この観察に基づき，各整数が約数として現れた位置を最後 2 回分だけ管理しながら整数列を左から右に走査することで，右端が現在位置であるような区間に対する答えが「最後から 2 回目に現れた位置が区間の左端より右にあるような整数の最大値」として得られる．1 点更新区間最大値取得のセグメント木を利用して管理し，上の答えを二分探索で取得することで，約数の個数の最大値を $\mu_{\max}$ として $\mathrm{O}((a_{\max} + n\mu_{\max} + q) \log a_{\max})$ 時間でこの問題を解ける．この問題では $a_{\max} \leq 10^5$ であり，$\mu_{\max} \leq 128$ であることが確認できるので，適切に実装することで制限時間に間に合わせることができる．

なお，区間の処理順を工夫して高速化する方法（いわゆる "Mo's algorithm"）や，約数全体を平方分割して出現位置を管理する方法などでも，定数倍に気を付けて実装すれば正解することは可能である．実際にそのような方法で正解したチームもいくつかあったようである．

J: Mixing Solutions

原案: 丸茂 直貴　主担当: 丸茂 直貴　（以下に記述する想定解法の 1 つは森田 晃平の提案）

防衛枠の最難問の 1 つである．昨年の問題 I と似た，液体を混合する問題である．今回の設定では，いくつかの水溶液を材料として目標である 1 つの水溶液を作りたいが，材料となる水溶液の濃度に不確実性がある．不確実性は濃度の区間として与えられ，その中であり得る最悪の状況における目標濃度 c との誤差が最小となるような混合方法を求めたい.（いわゆる「ロバスト最適化」の一例である.）

まず,「最悪の状況」とは，混合に用いた各水溶液の濃度が全て区間の下限と上限のいずれか一方に揃っている場合である．すなわち，下限に揃っていた場合の濃度を l，上限に揃っていた場合の濃度を r としたとき，目標量の水溶液であって $\max\{|l-c|, |r-c|\} = \max\{c-l, r-c\}$ を最小化するものを作りたい．ここで重要な観察として，$(l, r) \in \mathbb{R}^2$ の取り得る値全体は凸多角形をなす．なぜなら，実行可能な（総量が目標値 s に一致するような）混合方法が複数あったとき，それらの凸結合で表される混合方法は全て実行可能だからである．この観察から，$x \coloneqq c-l$, $y \coloneqq r-c$ と置き換え，$(x, y) \in \mathbb{R}^2$ の取り得る値全体（これも凸多角形となる）を P として，問題を以下の形に言い換えられる．

$$\underset{(x,y)\in P}{\text{minimize}} \quad \max\{x, y\}$$

凸多角形 P がどういう形をしているかを考えよう．まず，(x, y), (y, x) の辞書順最小を達成する頂点をそれぞれ (x_1, y_1), $(x_2, y_2) \in P$ とする.（これらは素朴な貪欲法で求められる.）このとき，$x_1 \geq y_1$ であれば答えは x_1 であり，$x_2 \leq y_2$ であれば答えは y_2 である．

以下では，$x_1 < y_1$ かつ $x_2 > y_2$ の場合を考える．このとき，最適解は直線 $y = x$ と凸多角形 P の交点（のうち左下にあるもの）であり，これを高速に求められればよい．P は凸多角形であるので，求めたい点は隣接する 2 つの頂点の凸結合として表せる．したがって，その 2 つの頂点が求まれば十分である．

一般に凸多角形の頂点は，あるベクトルとの内積の最小値を達成する唯一の点として特徴付けられる．ベクトル $v = (v_x, v_y) \in \mathbb{R}^2$ との内積を最小にするような P 内の点は，材料の水溶液を v との内積の値でソートして貪欲法を適用すれば求まる．さらに，候補となるベクトル v は，上記の内積の値でのソート結果が異なるものだけでよいので，$\mathrm{O}(n^2)$ 通りを試せば十分である．これらのベクトルは，$v = (0, 1)$ から始めて $v = (1, 0)$ に至るまで，ソート結果で隣接ペアが次に入れ替わるタイミング（v_x/v_y の値）を優先度付きキューで管理しながらソート結果を素朴に更新していけば，$\mathrm{O}(n^2 \log n)$ 時間で列挙できる．列挙した $\mathrm{O}(n^2)$ 個のベクトルを用いて二分探索により所望の 2 つの頂点を見つけることで，全体として $\mathrm{O}(n^2 \log n)$ 時間でこの問題を解くことができる．

なお，$\max\{x, y\} = \max_{t\in[0,1]}(tx + (1-t)y)$ と変形した目的関数の双線形性を利用し，minimax 定理を用いて min と max を入れ替えて整理するなどの方法でも同様の計算量で解ける．

K: Scheduling Two Meetings

原案: 稲葉 一浩　　主担当: 稲葉 一浩

m (≤ 20) 人での会議を 2 回開きたく，n $(\leq 2 \times 10^5)$ 個の日時候補と各人の出席可能性が与えられるので，全員が 1 回は出席できるような日時設定であって，のべ出席人数が最大となるようなもの（のうち日時が辞書順で最小のもの）を求めよ，という問題である．愚直に全てのパターンを確認すれば $\Theta(mn^2)$ 時間で正しく答えることができるが，もちろんこれでは間に合わない．この問題では，n は大きいが m が小さいので，これを利用した解法を考えよう．1 回目の日時候補に対して，2 回目は 1 回目の補集合が出席できる日時候補のうち出席者数が最大のもの（のうち最も早いもの）のみが候補となり得る．したがって,「各集合に対して，その中の全員が出席可能な日時候補のうち出席者数が最大のもの（のうち最も早いもの）」を高速に前計算できれば，あとは愚直に n 回確認すればよい．この前計算は，高速ゼータ変換の要領で m 次元の累積最大値を計算することで $\Theta(n + m2^m)$ 時間ででき，この問題に正解できる．ただし，全員出席可能な日時候補がある場合はコーナーケースであり，これだけは別途何らかの方法で処理する必要がある．

余談であるが，今年の審判団のメンバーは（審判長を除いて）ちょうど 20 人であり，実態に即した問題設定であった．ただし，日時候補が 2×10^5 個もあったことは未だかつてない．

L: Peculiar Protocol

原案: 稲葉 一浩　　主担当: 隈部 壮

正整数列と整数 d, r $(d \geq 2,\ 0 \leq r < d)$ が与えられ,「和を d で割った余りが r であるような区間を抜き出してその商の得点を得る（その後抜き出した部分は詰める)」という操作を可能な限り繰り返して合計得点を最大化する問題である．整数列の長さが $n \leq 500$ なので，いかにも $\mathrm{O}(n^3)$ 時間の区間 DP（動的計画法）を設計したくなる見た目であり，実際それは正しい方針である．ただし，直接答えを求めにいくのは（おそらく）上手くいかず，中間的な問題を挟む必要があるのが面白いポイントである．まず部分問題として，各区間について「その部分を最終的に全て抜き出したときに得られる合計得点の最大値」を計算しよう．これが計算できれば，元の問題の答えは素朴な $\mathrm{O}(n^3)$ 時間の区間 DP で計算できる．

部分問題では，各区間について「その部分を最終的に全て抜き出すための操作回数の最小値」が計算できればよい．なぜなら，区間内の整数の合計値は一定であり，合計得点はその合計値から操作回数の r 倍を引いて d で割った値に等しいからである．ここで，b を求めたい最小値とすると，b は「合計値と br を d で割った余りが等しい」ような最小の正整数である．そうでないとすると，最後の複数回の操作をまとめて 1 回でやれるはずで，b の最小性に矛盾するからである．さらに，最終的に区間全体を抜き出す操作ができるかどうかは，その（真に）内側でできる操作回数のみから決まり，各区間内でできる操作回数は明らかに連続的（0 以上最大値以下）である．これらの観察に基づき，各区間について「その中でできる操作回数の最大値」を $\mathrm{O}(n^3)$ 時間の区間 DP で計算し，その値が上で定義した b 以上であるかを確認することで，部分問題を解くことができる．

クエリデータ構造系問題の I とともに，AtCoder で育った人々（どれだけ居るかは定かではないが，少なくとも審判長を含む）にとっては，典型寄りで D より取り組みやすい問題ではないかと想定していたが，本コンテストでは D に比べてあまり解かれない結果となった．ミラーコンテストでは D, L を解いたチーム数がそれぞれ 12, 11 であり，難易度的に大きな差があるわけではなく，初動によるバイアスが大きかったものと考えられる．