

2014 年度は 11 問を出題した。正解数ごとのチーム数および、問題ごとの submit 数と正解チーム数は、表 1 および表 2 に示すとおりとなった。今回は全チームが 3 問以上解いたが、これは国内で開催した大会では初めてのことだと思う。全問正解したチームはなく、正解数は 3-10 問の範囲で平均的に分布していることからみて、問題セット全体の難易度設定はおおむね適切だったと言えるだろう。全チームが正解した問題 A, B, C も例年と比べて易しすぎる問題ではなく、出場チームの実力が高くなっていることを再確認させられる結果になった。

昨年までは問題数を 10 問とすることが多かったが、今年度はコンテスト時間が 5 時間と変わらないまま問題数を 11 問に増やした。問題作成の手間は増したが、新しく加わった審判を中心に、これまでになくタイプの問題が作られ、分野のバリエーションに富んだ問題セットになったと思っている。

表 1: 正解数ごとのチーム数

正解数	11	10	9	8	7	6	5	4	3	2	1	0
チーム数	0	1	2	1	5	7	6	9	9	0	0	0

2: 問題ごとの結果(全 40 チーム)

問題	分野	Submit 数	正解チーム数
A	整列	46	40
B	構文解析	47	40
C	貪欲法	50	40
D	力学計算	49	21
E	シミュレーション	36	11
F	グラフ	78	31
G	データ構造	59	15
H	幾何	5	3
I	ゲーム	15	3
J	最適化	11	2
K	動的計画法	0	0

以下では問題ごとの簡単な解説を記す。

問題 A. Bit String Reordering

当初は縦棒だけのあみだくじに最小個数の横棒を加えて目的の順列を作る問題として提案されたが、海外チームが不利になる可能性があったため、文字列の隣同士の文字を交換する問題となった。

想定解法はターゲット文字列の各位置の文字が初期文字列のどの位置に出現していたかを求めて、隣同士の文字を交換する整列法での交換回数を数える(あるいは、それと同等の概念である反転数を計算する)というものだが、文字の種類が 0, 1 の 2 値、文字列長の上限が 15 と状態空間が十分小さいので、幅優先探索で解いたチームも多かった。最初の正解が submit されたのは、競技 7 分経過後だった。

時間をかければ解ける問題であり、最終的には全 40 チームが正解した。

問題 B. Miscalculation

1 桁の数字と演算子「+」、「*」だけからなる式を、「乗算を先に」、「左から順に」の 2 種類の計算法で計算する問題である。構文解析の問題ではあるが、構文解析に関する知識はほとんど必要ない。

「左から順に」の実装は自明である。「乗算を先に」の方はスタックを使う方法が自然だが、スタックサイズが 2 で十分なので、変数で表してもよいし、1 パス目で乗算だけを計算し、2 パス目で加算を計算する方

法, 文字列の置き換えで計算を進める方法などさまざまな解法を許す問題になっている。
さまざまな解法でのプログラムの提出があり, 最終的には全 40 チームが正解した。

問題 C. Shopping

一本道にあるすべての店を訪問して通り抜ける。ある店の後に別の(より入口に近い)店を訪問しなくては
いけないという制約が複数ある時に, 移動距離を最小にするという問題である。

探索問題だと判断して, 各制約をどの順番で満たすかを探索すると, 制約の数(最大 500)の階乗に比例
する計算時間が必要になる。逆行しなければいけない区間が重なっている時に, 区間の union を取って
いくと, すべての制約を満たす移動経路が自動的に得られることに気がつけば, 区間の union を取るだけ
の問題として解くことができる。

プログラミング自体は難しくないが, 簡単な解法があることに気がつかないチームがいるかもしれないと考
えていたが, 最終的には全 40 チームが正解した。

問題 D. Space Golf

重力のある空間で障害物に当たらないように, なるべく低速度で弾丸を発射して, 目的地に到達させると
いう問題である。バウンドを許しているが, 弾性衝突なのでバウンドの回数を決めると, どの位置でバウン
ドさせるのが一意に決まる。

物理の問題にしないために, プログラムを作成するのに必要な式はすべて与えたが, 問題文が長くなり敬
遠したチームもあったかもしれない。各物体をぎりぎりにかすめる軌道を選ぶのが常に最善であると誤解
していると思われる回答もあったが, 同じ速度で最も遠くまで飛ぶ 45 度の角度で発射して物体に当たら
ない場合は, そちらの方がぎりぎりにかすめる場合よりも遅くなる。

この問題は 21 チームが正解した。

問題 E. Automotive Navigation

格子座標の世界で道路網の地図が与えられている。道の上のある座標から自動車がスタートして, 各ス
テップで前ステップからの移動距離と現在の車の向きが観測される。最後のステップで, どの座標にいる
可能性があるかを答える問題である。

1 ステップごとにシミュレーションを繰り返して, 可能な状態を更新していくという解法で解くことが可能であ
る。ただ, 自動車がちょうど曲がる場合は, 曲がる前と後の 2 種類の方向のどちらが観測されるか分からな
いというところが問題を難しくしている。

複雑なデータの管理をいかにして間違いなく行かせるかを問う問題であり, 複雑さの壁を突破できなかったチ
ームが多かったようだ。正解は 11 チームにとどまった。

問題 F. There is No Alternative

与えられた連結グラフのすべての最小全域木に含まれる辺を求める問題である。

最小全域木を 1 つ求めてから, それを構成する辺を 1 つずつ取り除いて同じコストの最小全域木を構成
できるかチェックするという方針で構わないが, Kruskal

のアルゴリズムで毎回すべての辺をソートするような効率の悪いプログラムは制限時間に終わらないような
時間設定になっている。

方針が立てやすいせいか, 31 チームが正解した。

問題 G. Flipping Parentheses

開き括弧と閉じ括弧の対応が取れている長さ N の文字列が与えられて, 指定された箇所の括弧の向きを
反転させた時に, なるべく左にある 1 文字の括弧の向きを反転させて, 対応が取れた状態に戻し, どの位
置にある文字を反転させるかを求める問題である。同様の質問を全部で Q 回繰り返すことになっている。

実行時間のオーダーが $O(NQ)$ のプログラムを書くのは容易だが, これでは時間制限内に終わらない。 $O(Q \log N)$ とするために, データ構造を工夫してセグメント木を実装するのが想定解になっている。

セグメント木を使うことが分かっても、適切に実装するには手間がかかる問題だが 15 チームが正解した。

問題 H. Cornering at Poles

平面上で円形のロボットがポールを避けながら、最短経路を通って目的地に到達するという問題である。ロボットを大きさのない点に変えて、ポールを円形の障害物にしても同じなのでそちらに置き換えた方が考えやすいだろう。最短経路が円に接する線分と円の弧の列から構成されることに気がつけばグラフ上の最短経路問題に帰着することが可能である。しかし、線分や円弧が他の障害物と交差ししないかどうかの幾何的な判定をミスなく実装しなくてはならない。

時間をかければ解ける問題だが、手をつけなかったチームが多かったと思われ、正解は 3 チームにとどまった。

問題 I. Sweet War

2 人ゲームの勝ち負けを決める問題である。ゲーム途中のすべての状態に勝負をつければ良く、体力のペアではなく体力の差を状態とすることにより、状態数を大幅に削減できること、手番のプレイヤーの体力が相手の体力以下の時にパスするのは意味がないことまではすぐに気がつくと思う。ただ、体力の値が 10^9 までとなっているので、体力の差ごとに得られる満足度をテーブルにする動的計画法ではメモリが足りなくなる。満足度の総和が 150 以下の正の整数であることを利用して、満足度を達成するのに必要な最低の体力差をテーブルにして動的計画法を適用すれば良いことがわかる。

方針が定まれば書くべきコードの量は多くはないが、手番により視点の変わる 2 人ゲームの実装には慣れていないチームが多いのか、正解は 3 チームにとどまった。

問題 J. Exhibition

n 個の 3 つ組の中から k 個を取り出し 3 つ組の要素ごとの和を計算し、この 3 つの値の積を求める。この値を最小化する k 個の組を求めて、その後で、特定の 3 つ組の数字を与えられたペナルティで書き換えて、 k 個の中に入れるという問題である。本質的に難しいのは前半部分なので、この部分だけを以下で説明する。

n 個から k 個を取り出す組み合わせをすべて作るのは実行時間が不足し、最適化問題としてみると、各要素の和の積の最適化は難しいことがわかる。そこで、各要素の和の積が最小となる k 個の要素からなる集合は、ある重みに関して各要素の和の重み付き和も最小とするという性質を用いることにする。各要素の和の重み付き和を最小にする k 個の取り出し方は $O(n^k)$ 通りなので、すべて生成した上で積が最小となる組み合わせを求めれば良い。

説明されれば理解できるものの解答の場で思いつくにはかなりの数学的センスを必要とする難問だが、2 チームが正解した。

問題 K. L_∞ Jumps

格子座標系で、ある点から L_∞ 距離が一定の点へのジャンプを指定した回数繰り返して目的の点に至る。各回にどの相対位置にジャンプするかに応じてコストが指定されて、コストを最小化するという問題である。座標の範囲が 10^{10} と大きいので、座標をテーブルのインデックスとする単純な動的計画法では無理だと分かるが、コストの与え方の規則性に気がつく状態数を減らした動的計画法で解けることがわかる。

かなりの難問で残念ながら正解チームはいなかった。