審判長講評

審判長　鴨 浩靖

全体を見て

表1 に，今回の結果をまとめた．さらに細かく見るために，表2 に，問題ごとの結果をまとめた．

歴代の審判団は，おおむね次のような目標を立てて作題してきた．

1. どのチームも1 問以上正解すること．
2. 全問正解チームがないこと．
3. どの問題も1 チーム以上が正解すること

今回は，この三つの目標をすべて達成することができた．

問題A, B, C は，いずれも一般的なプログラミング教育の初級レベルまたは中級レベルで学ぶことだけを使って解ける問題である．問題A, B は全チームが，問題C は7 割のチームが解いている．アジア地区予選に出場する選手は，基本的な知識と技能についてはとりこぼしはないようだ．

一方，幾何アルゴリズムを必要とする問題(D, E, I) は，いずれも正解チームが極端に少ない．挑戦したチーム数も少ないので，最初からあきらめたチームが多かったのだろう．日本には幾何に苦手意識を持つ選手が多いと聞くが，上位を狙うには，幾何の易しい問題をとりこぼさないことが大切だ．

問題F, H, I, J は，いずれも，該当分野の基本的なアルゴリズムの知識があれば，実装はややこしくはない．問題F, H は半数のチームが解いたが，問題I, J は1 割前後のチームしか解けなかった．これだけから，グラフ理論(問題F) と形式言語理論(問題H) は得意とする学生が多いが，計算幾何学(問題I) や代数学(問題J) はそうでないと断定するのは論理の飛躍だろうが，いずれにせよ，計算幾何学も代数学もソフトウェア研究者／技術者にとって，重要な分野ではあるので，これを機会に深く学んでほしい．

表1: 正解数ごとのチーム数(全10 問)

正解数	チーム数
10	0
9	1
8	3
7	4
6	5
5	6
4	14
3	5
2	12
1	0
0	0

表2: 問題ごとの結果(全50 チーム)

問題	分野	挑戦したチーム数	正解したチーム数
A	シミュレーション(整数)	50	50
B	整数	50	50
C	組み合わせ	40	35
D	幾何	3	2
E	幾何・グラフ	4	3
F	グラフ	28	25
G	探索	17	16
H	言語	28	26
I	幾何	11	7
J	台数	2	2

問題ごと

問題A　And Then There Was One　環状に並べた石を指定された個数おきに除いていって，最後に残る石はどれかを答える問題である．

古くから知られる問題で，江戸時代を通じたベストセラーであった数学書『塵劫記』にも，「継子立て」という名で類題が掲載されている．初等整数論を利用した解法もあるが，この問題の規模(石は最大で 10000 個) ならば，石を並べて取り除いていくことをシミュレートする方法で解ける．石の並びを配列かリストを使って実装し，その要素を数えて削除するだけでよい．

基本的なデータ構造が扱えさえすれば解ける問題であり，全チームが正解したのは順当な結果だろう．

問題B　Prime Gap　与えられた正整数が含まれる素数砂漠(素数と素数にはさまれた合成数の並び)の長さを求める問題である．

与えられた正整数以上で最小の素数と，以下で最大の素数を求め，その差を計算すればよい．素数を求めるには，エラトステネスの篩によって素数の表をあらかじめ作っておく方法が一般的である．この問題の規模では，2 以上$\sqrt{n}$ 以下の整数で順に割っていく素朴な素数判定法でも，十分に間に合う．

プログラミングを学ぶ学生のほとんどは，初歩の段階でエラトステネスの篩や素朴な素数判定法を学んでいるだろう．ACM ICPC でも，素数に関わる問題は過去に多く出題されているので，コードを用意していたチームも多かっただろう．この問題を全チームが正解したのは，順当な結果だろう．

問題C　Minimal Backgammon　双六に似た一人遊びで，決められた手数以内でゴールできる確率を求める問題である．

サイコロの目に応じて駒の動きを追っていって，ゴールできる場合の数を数えれば良いのだが，素朴なアルゴリズムでは，制限時間を超過してしまう．動的計画法かメモ化を用いて，同じ計算の繰り返しを防ぐ必要がある．

この問題は易しい問題ではあるが，中級者程度の知識は必要であり，初級を学び終えた程度の知識では解けない．7 割のチームが正解したのは，順当な結果だろう．

問題D　Lowest Pyramid　座標平面上に三角形(の頂点の座標)が与えられる．側面となる三つの三角形を加えて，それを底面とする三角錐の展開図を作る．ただし，与えられた三角形も新たに加える三角形も，その頂点の座標は方眼の格子上にある．その条件をみたして作られる三角錐の高さの最小値を求める問題である．

この問題を解くアルゴリズムは何通りかあるが，いずれも，頂点の候補を選び，三角錐が作れるかを調べ，作れる場合の高さを求めることを繰り返し，その最小値を求めるものである．どうやって頂点の候補を効率良く絞り込むかがポイントである．

この問題を2 チームしか正解しなかったのは，少なすぎる．正解には達しなかったチームを含めても3 チームしか挑戦していないことから，最初から難しいものと思いこんで手を出さなかったチームが多かったのではなかろうか．もう少し，多くのチームが挑戦して正解してほしかった．

問題E　Geometric Map　地図を読み，二地点間の最短路を求める問題である．地図は線分の集まりとして与えられるが，その一部は通りを表し，他は一方通行を表わす．

線分の長さと互いの位置関係を読みとるためには，平面上の線分の長さや角度や交点を求めるアルゴリズムを組合せれば良い．最短路を求めには，たとえば，ダイクストラ法を使えば良い．

この問題の正解チームが極端に少なかったのは，解法が幾何的な部分とグラフ的な部分の二段階になるので，手を出しにくかったためだろう．正解した2 チームは，優れたコーディング力を持っているといえよう．

問題F　Slim Span　重みつき無向グラフの全域木のうちで，もっとも重い辺ともっとも軽い辺の重みの差が最小となるものを求める問題である．

最小全域木（全域木のうちで辺の重みの和が最小となるもの）を求めるアルゴリズムは，複数のものがすでに良く知られている．そのどれかをひとひねりすることで，この問題を解くアルゴリズムを得ることができる．

これは，グラフアルゴリズムの理解が問われる問題である．半数のチームがこの問題を解けたことは，グラフアルゴリズムを理解している学生がそれだけ多いということであり，喜ばしい．

問題G　The Morning after Halloween　複数のロボットを動かして決められた配置にする最短の手数を求める問題である．

幅優先探索で実現するのが，実用的なほとんど唯一の方法であろう．

使用するアルゴリズムは難しいものではないが，条件が複雑な分だけコードも複雑になる．この問題を解くには，条件を正確に読みとり，確実に実装する能力が問われる．16 チームしか正解できなかったのは，実装に手間取って時間切れになったチームが多かったからだろうか．

問題H　Bug Hunt　配列宣言と代入のみをもつとても単純なプログラミング言語のプログラムが与えられ，それを実行したときに実行時エラーを起こす個所をみつける問題である．

簡単な構文解析系と簡単なインタプリタを実装すれば良い．

この問題も約半数のチームが正解している．言語処理の知識が十分に普及しているということであり，喜ばしい．

問題I　Most Distant Point from the Sea　凸多角形の内部で，周から最も遠い点を求める問題である．

解は，二辺（隣り合わなくても良い）が作る角の二等分線の交点であるから，そのような点をすべて求め，そのうちもっとも適切なものを選ぶ，解法がありえる．ただし，この解法は，二辺が平行な場合の例外処理など，繁雑なプログラミングが必要になる．

多角形から一定の幅で削った図形が空であるか判定するアルゴリズムを使って，二分探索で近似する解法も可能である．プログラミングが最も簡単なのは，この方法だろう．

この問題を，7 チームしか正解できなかったのは残念だ．計算幾何の基本的なアルゴリズムを知っていれば解ける問題なので，もっと多くのチームが正解してもよかったように思う．幾何は難しいという先入観があるのだろうか．

問題J　The Teacher’s Side of Math　$\sqrt[m]{a}+\sqrt[n]{b}$（ただし，a とb は相異なる素数で，m とn は2 以上の整数）を根にもつ整数係数多項式方程式を求める問題である．

多項式の係数がみたすべき連立一次方程式を求め，ガウスの消去法でそれを解くことが，標準的な解法である．

代数学の知識があれば簡単だが，知識がなければ手も足も出ない問題である．2 チームしか解けなかったのもしかたないだろう．なお，この問題に挑戦したチームがすべて正解しているのは，アルゴリズムさえわかれば実装は簡単であることを反映しているのだろう．