SuperCon2007 課題

「星域管理基地配置問題」

1　あらすじ

すでに人類が宇宙に進出している紀元 2407 年、人類は超光速通信手段「アンシブル」を開発した。これにより、遠く離れた星の間での即時通信が可能となり、相対論的限界を超える広い領域を管理できるようになったのである。

さて、2507 年、新たに人類の領土となったある星域を管理するために、いくつかの星に基地を配置する必要が生じた。各基地はアンシブル・トランスミッタブル・フィールド (通称 AT フィールド) と呼ばれる場を形成し、そのフィールドに含まれる星とのあいだではアンシブルを用いた即時通信を行なうことができる。

星は宇宙空間に散らばっているが、均等に散らばっているわけではなく、比較的多くの星が集まる領域もあれば、星がまばらな領域もある。コストを考えると多くの星が集まる領域に基地を配置したい。

そこで、なるべく多くの星が管理下に置かれるよう、基地をうまく配置しようというのが問題である。

しかし、ふたつ以上の AT フィールドが重なると干渉によってフィールドが消失してしまうという問題がある。AT フィールドは基地を中心とした一定半径の球を成すので、各基地はその半径内に含まれる星しか管理できない。そのため、残念ながらどの基地の管理下にもない星ができてしまい、それが有名な 2607 年の反乱へとつながることになるのだが、それはまた別の話である。

なお、アンシブルの開発経緯についてはアーシュラ・K・ル・グインの『所有せざる人々』に詳しいが、そこに AT フィールドについての記述があるかどうかは定かではない

2　問題の詳細

3 次元空間中に撒かれた N 個の星 (0 番から $N-1$ 番まで) の座標の組 $(x_0, y_0, z_0) \sim (x_{N-1}, y_{N-1}, z_{N-1})$ とそれらを覆うべき球の半径 R が与えられる。星は大きさのない「点」と考える。N 個の星の中からいくつかを選んで基地とし、それらを中心とする半径 R の球を考える。このとき、できるだけ多くの星が球の内部に含まれるように、基地をうまく配置せよ。

基地はいくつ配置してもよいが、球同士は重ならないものとする。なお、球には、中心となる星 (基地) 以外に最低 1 個の星を含むこと。

N の最大値は 30000 とする。また、各座標は 0 ～ 10000 の範囲の整数 (0 と 10000 も含む)3 個の組で与えられる。球の半径 R も整数で与えられる。この際、(1) 球の表面上に乗った星は球に「含まれる点」に数える (2) 球が一点で接する場合 (基地の中心同士がちょうど $2R$ 離れているとき) も「重なる」と考える。

解としては、使用する基地の個数と基地となる星の番号を答える。

3　勝利条件

制限時間内であれば、解をいくつ出力してもよく、最後に出力された解を採用する。制限時間に達しても計算が終了しない場合は強制終了させるが、それ以前に解が出力されていれば、それを採

用する。なお、制限時間は CPU 時間ではなく、実行時間である。

順位は、球に含まれずに残る星の数が少ない順とする。なお、球が重なった場合は条件に反するので、その球 (重なる二個両方) はなかったものとする。つまり、その球に含まれる星は、「含まれずに残った星」として数えられる。また、残る星の数が同数のときは計算の所要時間 (プログラムの終了までの時間ではなく、最後の解が出力されるまでの時間) が短いほうを上位とする。基地の数は順位と関係しない。

まず、東京・大阪それぞれで、予選問題によって上位 4 組ずつを決める。次にそれら 8 組のプログラムで決勝問題を解き、優勝チームを決定する。

4　考え方のヒント

最も簡単な考えは「欲張り法」だろう。まずは、含む星の数が最大となるように基地をひとつ設定する。次に残った星の中から、やはり含む星の数が最大となるように基地を設定する。これを条件を満たす星がなくなるまで続ける。これで少なくともひとつの解は出せるはずである。

「欲張り法」では必ずしも最良の解が得られるとは限らないので、その次に考えるべきなのは、「欲張り法」の条件を少し緩めてみることだろう。もちろん、「欲張り法」以外の解き方を工夫してみるのもよい。

5　プログラミング上の注意

問題データの入力と解の出力には、こちらで用意した関数 SC_init と SC_output を使う。それらの関数やその他、問題に使う定数や関数のうちいくつかはあらかじめヘッダーファイル sc07.h で定義されているので、それをインクルードすること。sc07.h はソースコードで提供するので、読むのはかまわないが、本番ではこちらの手許にあるソースを使うので、配られたものを修正してはならない。sc07.h 中に誤りを見つけたときは、申し出ること。

星の数 N は SC_NSTAR で定義する。30000 個以下のあらゆる星の数に対して、コンパイルしなおさずに対応できるプログラムを書くのではなく、星の数に応じて SC_NSTAR を定義し、そのたびにコンパイルしなおすというプログラムを書いてほしい。問題の規模によって必要とするメモリーサイズが大きく変わるが、計算機を効率よく使う観点から、計算に必要なだけのメモリーを確定させておくことが有効だからである。また、これにより、malloc を使う必要がなくなる。プログラム中でも星の数は SC_NSTAR で表記する。なお、SC_NSTAR は必ず、sc07.h をインクルードする前に定義する (sc07.h の中でも使うため)。以下は星の数 5000 個のヘッダー例である

```
#include<stdio.h>
#include<math.h>
#define SC_NSTAR 5000
#include "sc07.h"
```

SC_init は問題データのファイル名を引数とする。これを実行時の引数として渡せるように

```
int main(argc,argv)
int argc;
char *argv[];
```

```
{   いろいろ定義
SC_init(argv[1]);
```

などとする。SC_init は必ず、プログラムの最初の実行文とすること (宣言文中の初期値代入はかまわない)。これを実行すると、球の半径が整数変数 SC_Radius に、また星の座標が整数配列 SX_XStar に格納される。なお、SX_XStar に格納される要素は 0 番から SC_NSTAR-1 番までである。解答として星の番号が必要になるが、それはこの配列の要素番号で指定するものとする。

解答の出力はプログラム中で何度行ってもよい。出力の際は、基地となる星の番号を整数配列 SC_Base 中に格納し (m 個の基地を使うとすると、SC_Base[0] から SC_Base[m-1] までに格納。順番はソートしなくてよい。また、SC_Base[m] 以降には無関係なデータがはいっていてもよい)、SC_output 関数を実行する。関数の引数は基地の個数 m である。

```
SC_output(m);
```

ヘッダー中で各変数は以下のように定義されている。

```
int SC_Radius;
int SC_XStar[SC_NSTAR][3];
int SC_Base[SC_NSTAR];
```

ほかにもヘッダー中で変数が定義されているが、プログラム中で使うのは上の三つだけである。なお、使わないものも含め、すべての変数・関数には接頭辞 SC_がつけられている。

なお、星の数が N 個のとき、入力データは $N+1$ 行からなり、一行目に球の半径 R、2 行目以降の各行に各星の x, y, z 座標が書かれている。自分でデータを作ってみたいときは、このフォーマットで作ること。