

joi2012-sp-day2

テレビ放送 (Broadcasting)

Masaki Hara
@qnighy



2歳のとき強く頭を打ち、算数に目覚める。
9歳のとき強く頭を打ち、プログラミングに目覚める。
10歳のとき強く頭を打ち、Javaに目覚める。
14歳のとき強く頭を打ち、アニメに目覚める。
15歳のとき強く頭を打ち、アルゴリズムに目覚める。
17歳のとき強く頭を打ち、形式的定理証明に目覚める

<https://twitter.com/#!/qnighy/status/25315686027>

問題

- 平面上に N 個の点がある
- K 個の円を配置(場所・大きさは任意)し、それらの点をカバーする
- それぞれの円について、半径の二乗に比例するコストがかかる
- コストの合計値を小さくしたい

- 出力のみ提出

問題

- 平面上にN個の点がある
- K個の円を配置(場所・大きさは任意)し、それらの点をカバーする
- それぞれの円について、半径の二乗に比例するコストがかかる
- コストの合計値を小さくしたい

- 出力のみ提出

出力のみ問題！

- 一般的なアドバイスから

出力のみ問題！ (1)

- 入力を見ながらコーディングできる
 - 入力データの特徴は最大限活用しよう
 - そのためにビジュアライザが欲しい時もある

出力のみ問題！ (2)

- プログラムを提出する必要がない
 - いくつプログラムを使ってもよい
 - 手作業をしてもよい
 - 場合によっては有用(e.g. IOI2010 Maze)
 - ただし、Wrong Answerに注意

出力のみ問題！ (3)

- プログラムを提出する必要がない
 - 1秒以内に実行する必要はない
 - 長い時間かけて良い答えを得る
 - C/C++で書く必要もない
 - Ubuntuなら、bash, Perl, Pythonは確実に入っています
 - ブラウザでJavaScriptを動かすこともできます
 - ただし、コンテストの禁止事項には注意

出力のみ問題！ (4)

- 何回でも挑戦できます
 - 出力のみ問題では、手動でのパラメーター調整が重要になります。
 - 時間をかけるほど良い解が出るプログラムはgood
 - パラメーター調整が簡単にできるようにするとgood

出力のみ問題！ (5)

- ※5時間コンテストです！！
 - 出力のみ問題にこだわりすぎないように
 - 無制限に時間を消費するので、最後にするのが無難？
- 複雑な戦略(焼きなましやGA)は、5時間コンテストではあまり力を発揮しないかもしれません
 - パラメーター調整が難しいし。
 - 山登り法や「焼きっぱなし」がおすすりめです

出力のみ問題！

- 入力データの特徴を活用する
- いくつプログラムを作ってもよい
- 好きなだけ実行してよい



閑話休題

問題

- 平面上に N 個の点がある
- K 個の円を配置(場所・大きさは任意)し、それらの点をカバーする
- それぞれの円について、半径の二乗に比例するコストがかかる
- コストの合計値を小さくしたい

入力の性質

- ビジュアライザを書く
 - JavaScript+Canvasで絵を描けます

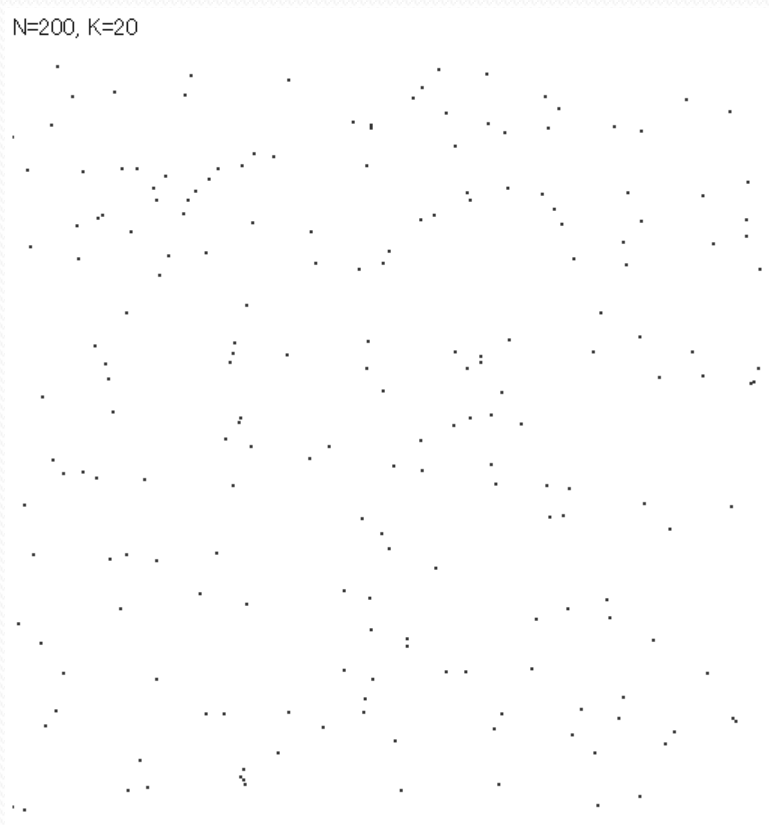
入力の性質

- ビジュアライザを書く

```
int N, K; fscanf(infile, "%d%d", &N, &K);<br>static int A[500], B[500];<br>for(int i = 0; i < N; i++) {<br>    fscanf(infile, "%d%d", A+i, B+i);<br>}<br>printf("<p>N=%d, K=%d</p>¥n", N, K);<br>printf("<canvas id='vis' width='%d' height='%d'></canvas>¥n", R, R);<br>puts("<script>");<br>puts("var vis = document.getElementById('vis');");<br>puts("var ctx = vis.getContext('2d');");<br>for(int i = 0; i < N; i++) {<br>    puts("ctx.beginPath()");<br>    printf("ctx.arc(%d, %d, 1, 0, 360);¥n", A[i]*R/1000000, B[i]*R/1000000);<br>    puts("ctx.fill()");<br>}<br>puts("</script>");<
```

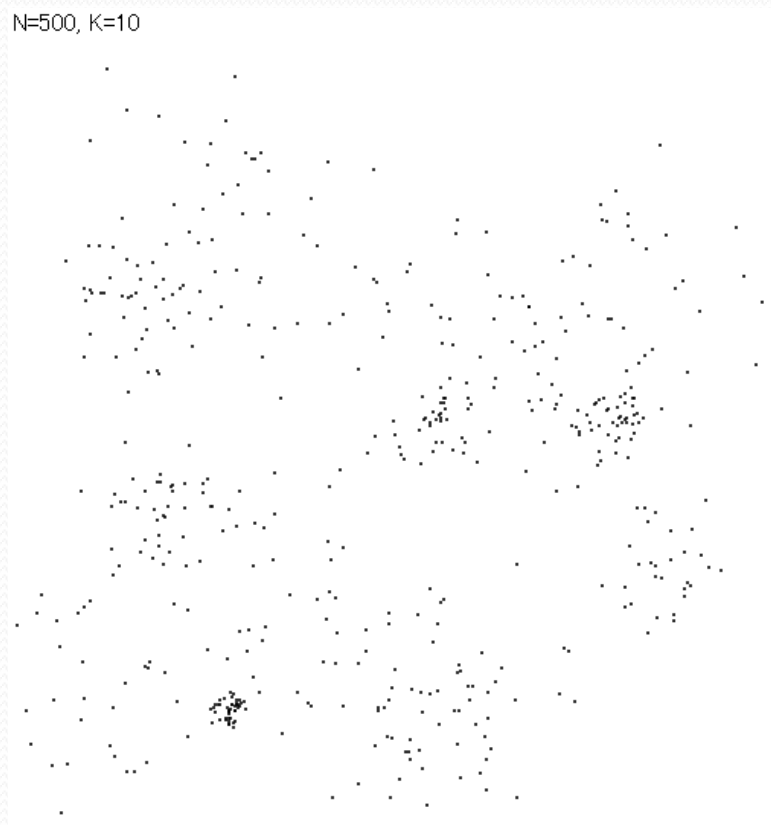
入力の性質

N=200, K=20



01.txt

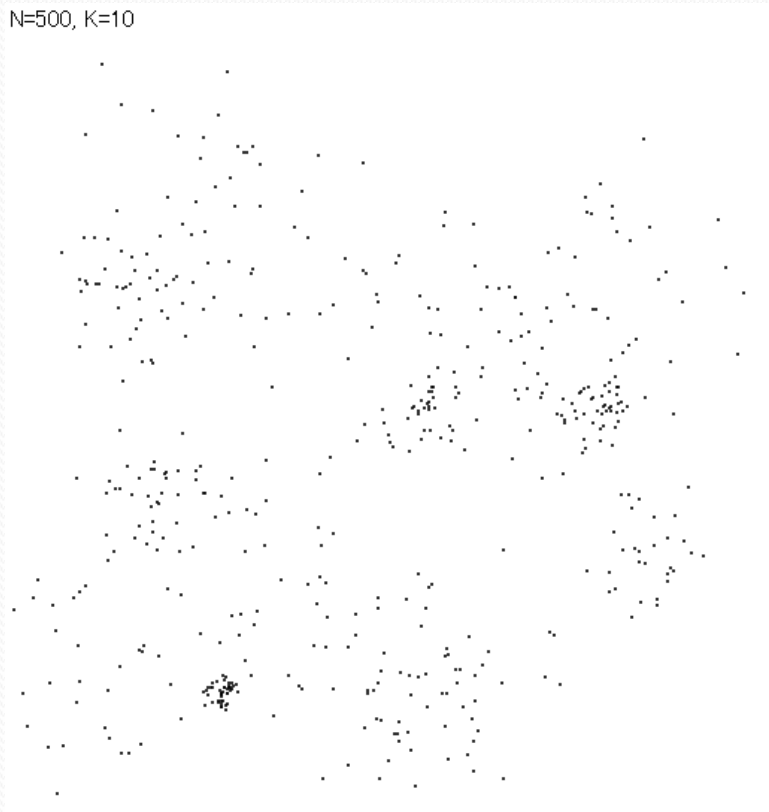
N=500, K=10



02.txt

入力の性質

N=500, K=10



03.txt

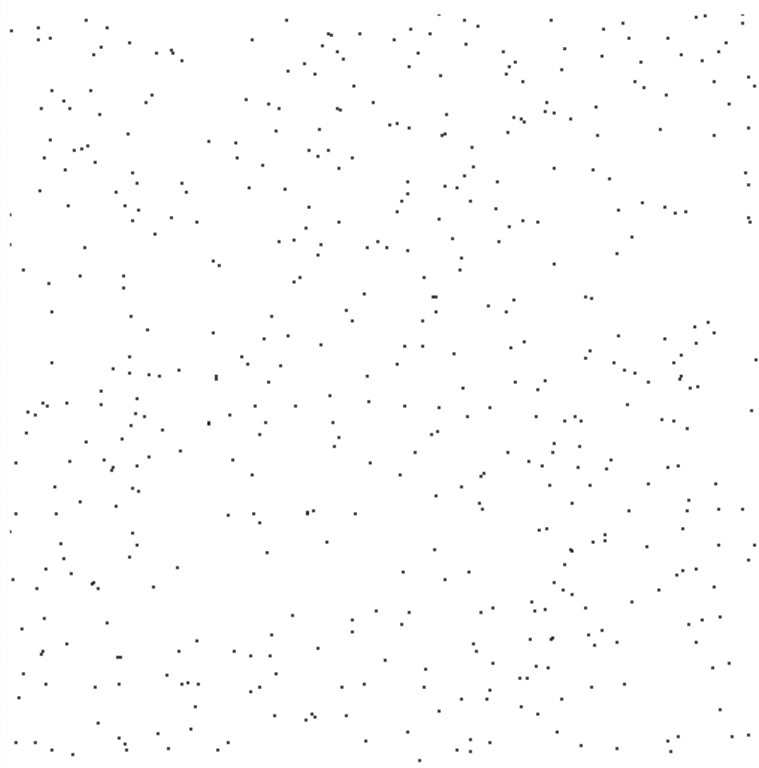
N=500, K=15



04.txt

入力の性質

N=500, K=30



05.txt

入力の性質: 考察

- 入力によって性質が違う
 - 均等なもの (01, 05)
 - 若干ばらつきのあるもの(02, 03)
 - 狙ってるとしか思えないもの(04)
- アルゴリズムによって得手不得手がありそう

考察

- 考察: カバーしたい点集合に対して、最小包含円を決めることができる
- →以下の2つの問題に分かれる
 - K個の点集合を作る
 - それぞれの点集合について、最小包含円を求める

最小包含円

- 与えられた点を全て含む円で、最も半径が小さいもの
- $O(n^3)$
- 平均 $O(n)$
- その他 - 貪欲法やエッセ最小包含円

最小包含円 - $O(n^3)$

- $n \leq 2$ のときは自明
- $n = 3$ のとき
 - 鈍角三角形の場合...一番長い辺が直径
 - 鋭角三角形の場合...外接円を考える
- $n \geq 3$ のとき
 - 全ての三角形を考えて、その最小包含円のうち最も大きいものを選べばよい

最小包含円 – 平均 $O(n)$

- $\text{min_disk}(P, Q)$: 点集合 P と Q を含む最小包含円を求める。
ただし Q は周上にあることが保証されている。
 - P が空なら、 Q の外接円が答え
 - P から点をひとつ選び、 p とする
 - p を除いた最小包含円 $\text{min_disk}(P - \{p\}, Q)$ を D とおく。
 - $p \in D$ なら、 D が答え
 - $p \notin D$ なら、 p は最小包含円の周上にある
 - $\rightarrow \text{min_disk}(P - \{p\}, Q + \{p\})$ が答え

最小包含円 – 平均 $O(n)$

- 点の順番がランダムになっているとする
- 最小包含円の周上に点は3個程度しかない
 - $\rightarrow \min_disk(P - \{p\}, Q + \{p\})$ が呼び出される確率は $(3 - \#Q)/n$ 程度
 - したがって、平均で $O(n)$

最小包含円 – その他の方法

- 中心を決めると、必要な半径は $O(n)$ で求まる
- 貪欲法
 - 円の中心を、必要な半径が小さくなる方向に進めていく
- 適当な値で代用する
 - 円の中心 = 全ての点をカバーする矩形の中心 などとおく
 - それなりに良く近似できるし、簡単に書ける

K個の点集合を決める

- 乱択で近傍探索
- クラスタ解析
- その他の方法

K個の点集合 – 近傍探索

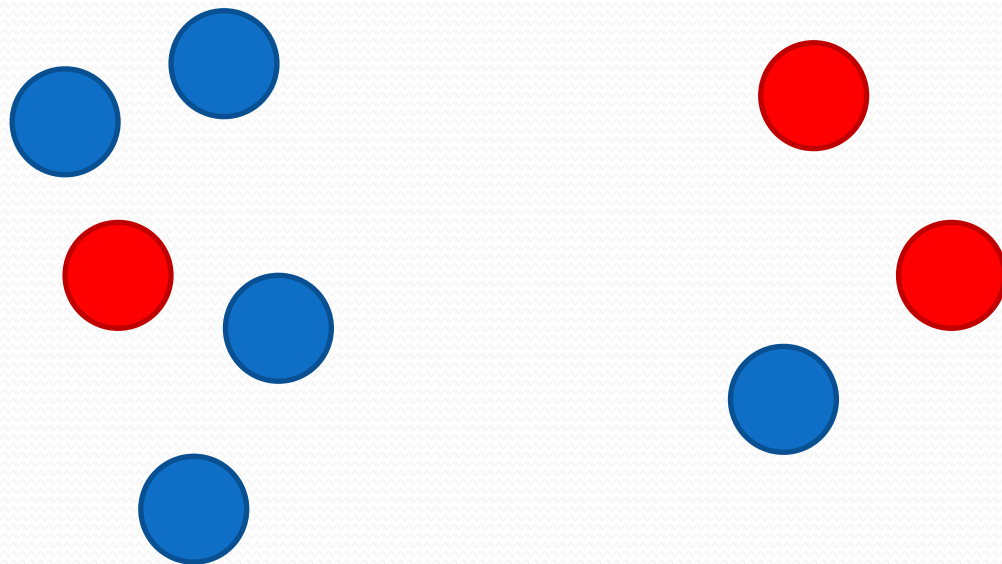
- とりあえず適当に点集合をおく(乱数とかで)
- 近傍は以下のように選ぶ
 - 適当な点を選び、適当な集合に移動させる
 - スコアに応じて、その近傍に移動する
- 均等なデータのほうが強い

K個の点集合 – クラスタ解析

- データ群を類似関係にあるグループに分類する方法
- ここでは「K平均法」を使う

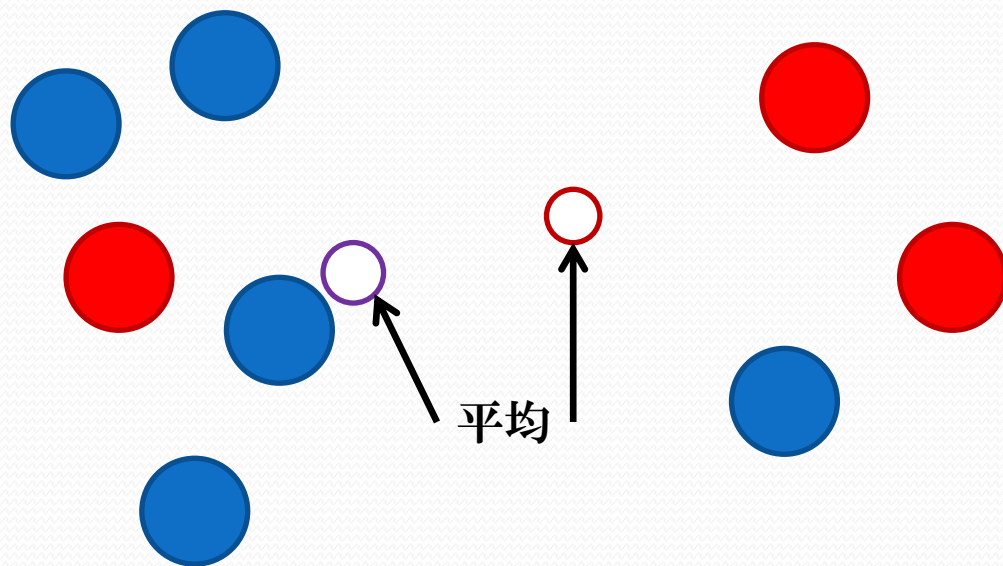
K個の点集合 – クラスタ解析

- データ群を類似関係にあるグループに分類する方法
- ここでは「K平均法」を使う



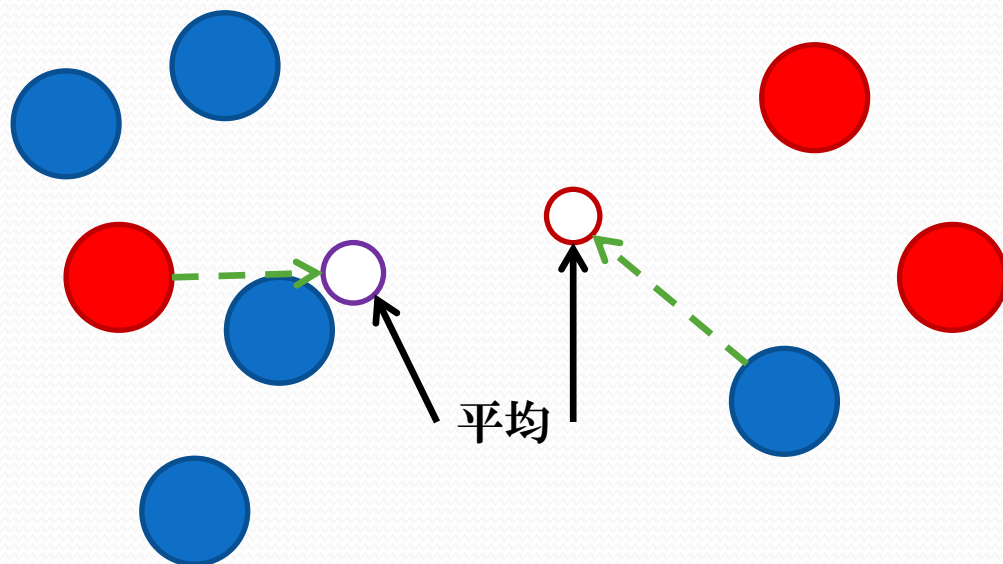
K個の点集合 – クラスタ解析

- データ群を類似関係にあるグループに分類する方法
- ここでは「K平均法」を使う



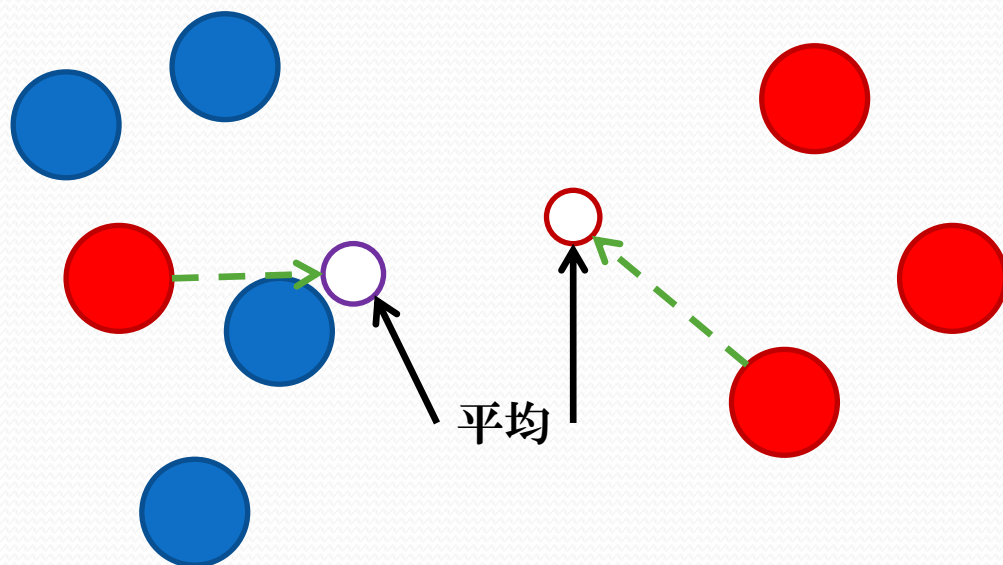
K個の点集合 – クラスタ解析

- データ群を類似関係にあるグループに分類する方法
- ここでは「K平均法」を使う



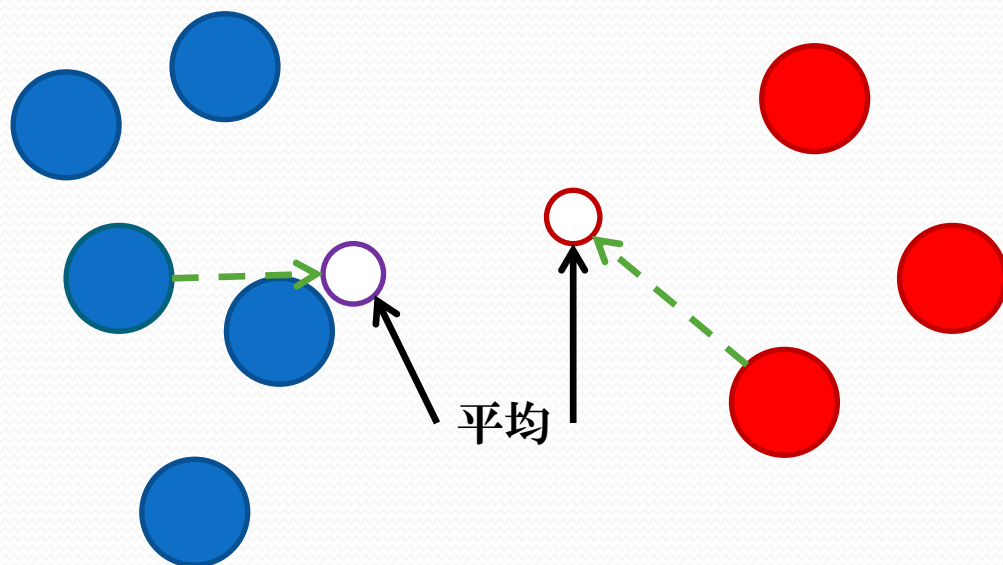
K個の点集合 – クラスタ解析

- データ群を類似関係にあるグループに分類する方法
- ここでは「K平均法」を使う



K個の点集合 – クラスタ解析

- データ群を類似関係にあるグループに分類する方法
- ここでは「K平均法」を使う



K個の点集合 – クラスタ解析

- データ群を類似関係にあるグループに分類する方法
- ここでは「K平均法」を使う

- まばらなデータに強い

K個の点集合 – その他の方法

- Kruskal法と同様に、N個から併合を繰り返してK個にする

その他の方法の紹介

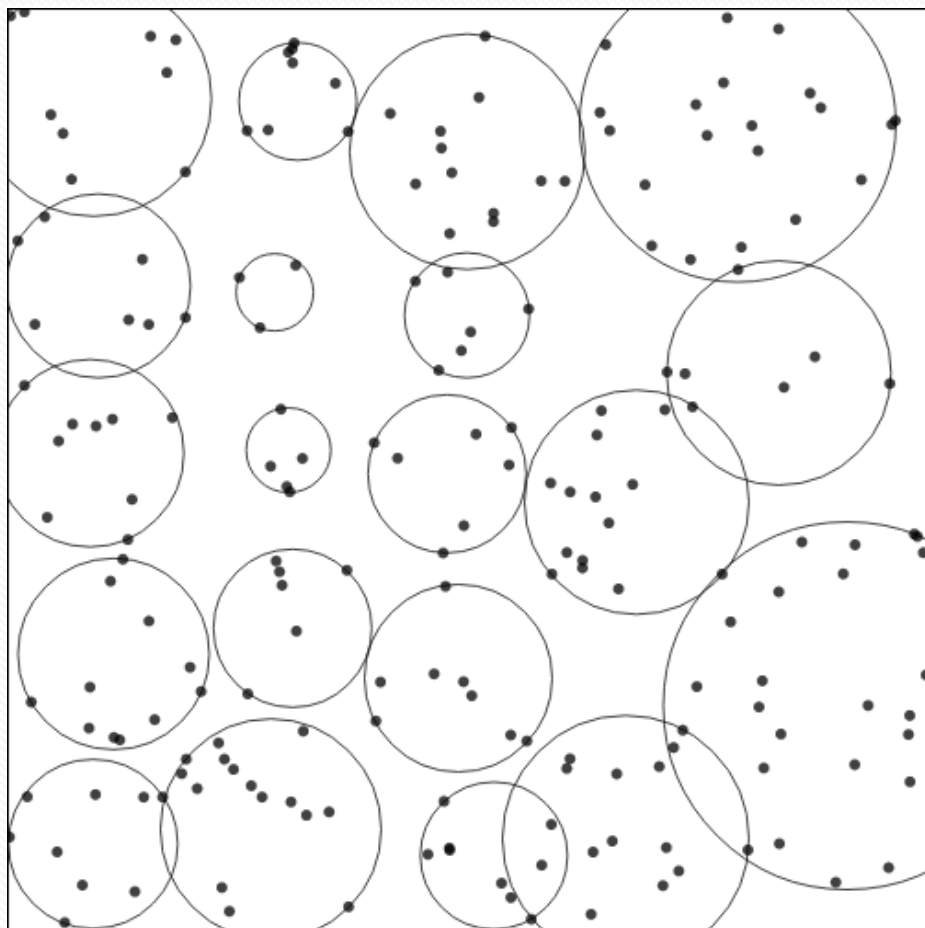
- 中心決め打ち
 - 貪欲法などを使って半径を決める



それでは

生徒の最良データの紹介

生徒の最良データの紹介

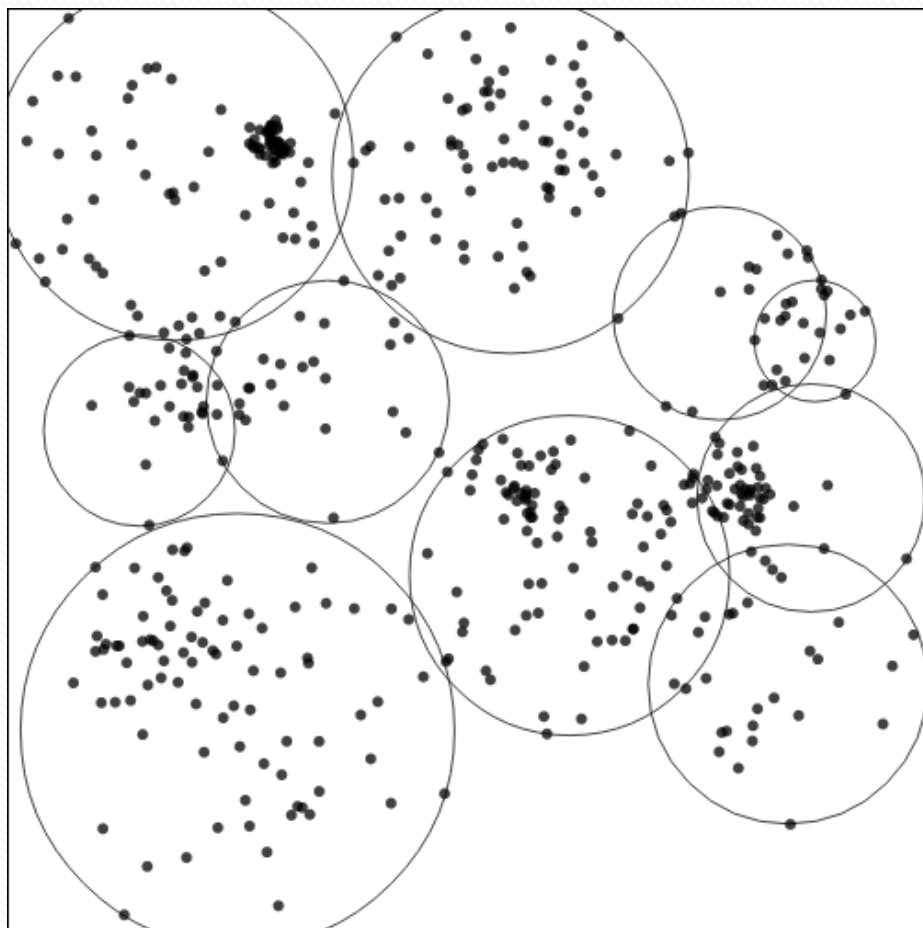


ino1

JPN17

スコア 247498181034

生徒の最良データの紹介

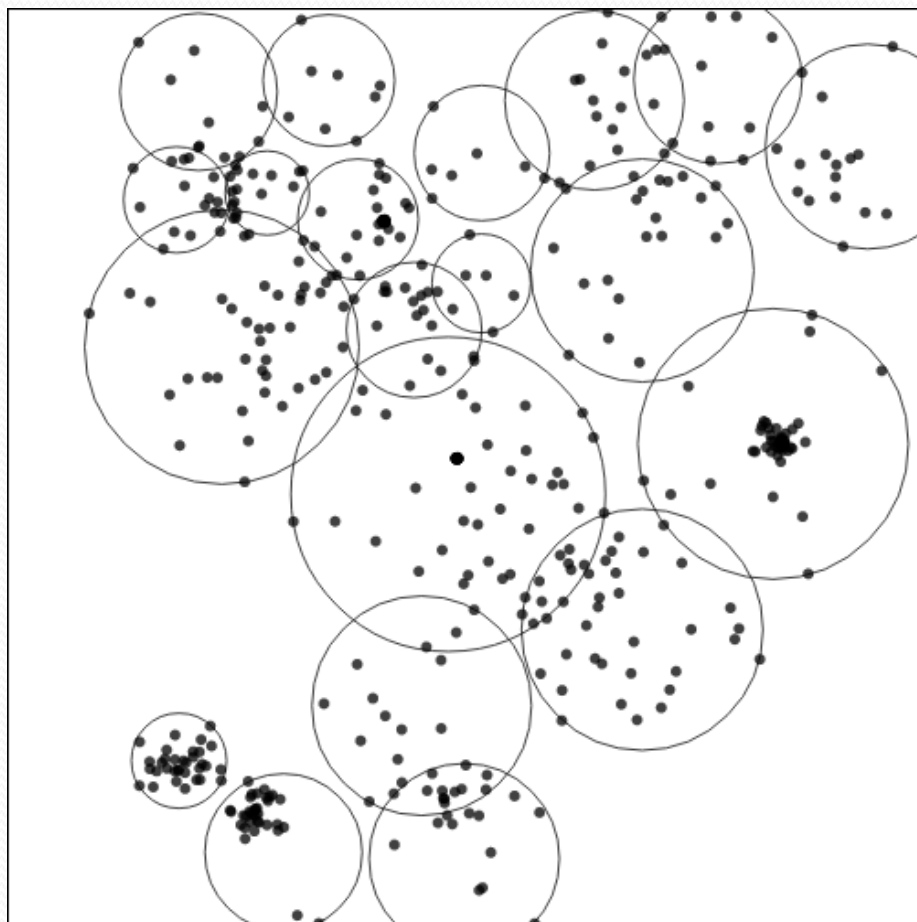


ino2

JPN18

スコア 246525708455

生徒の最良データの紹介

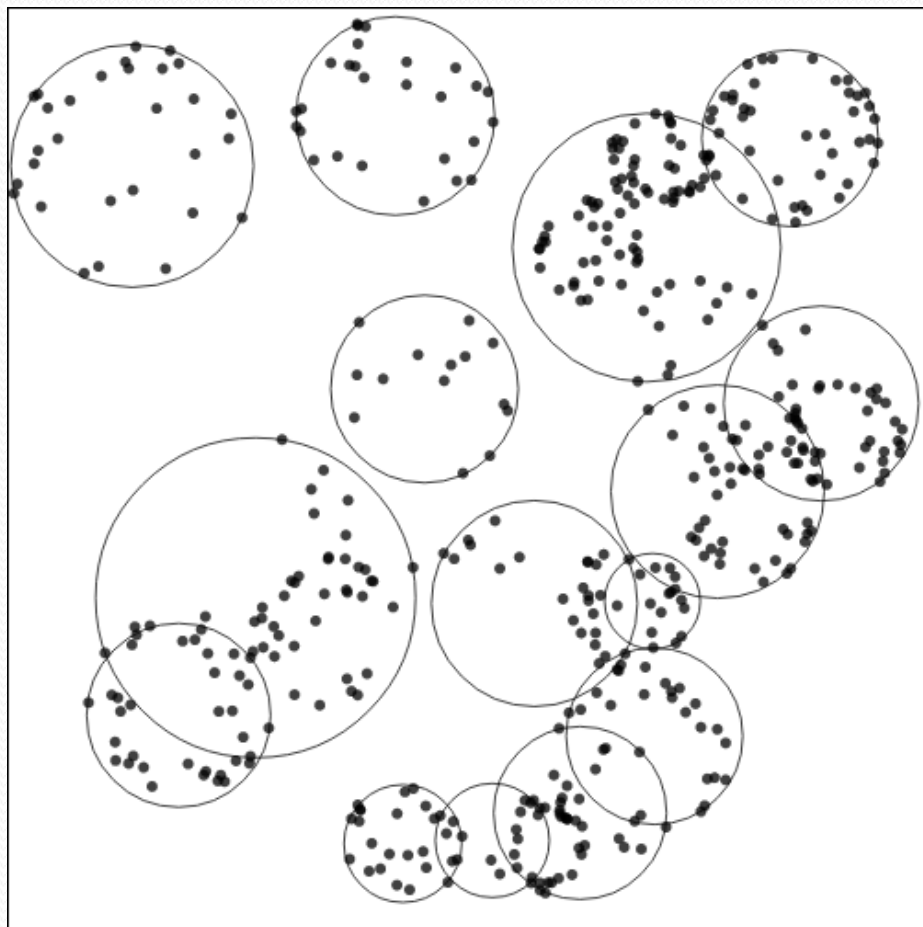


ino3

JPN18

スコア 208558463568

生徒の最良データの紹介

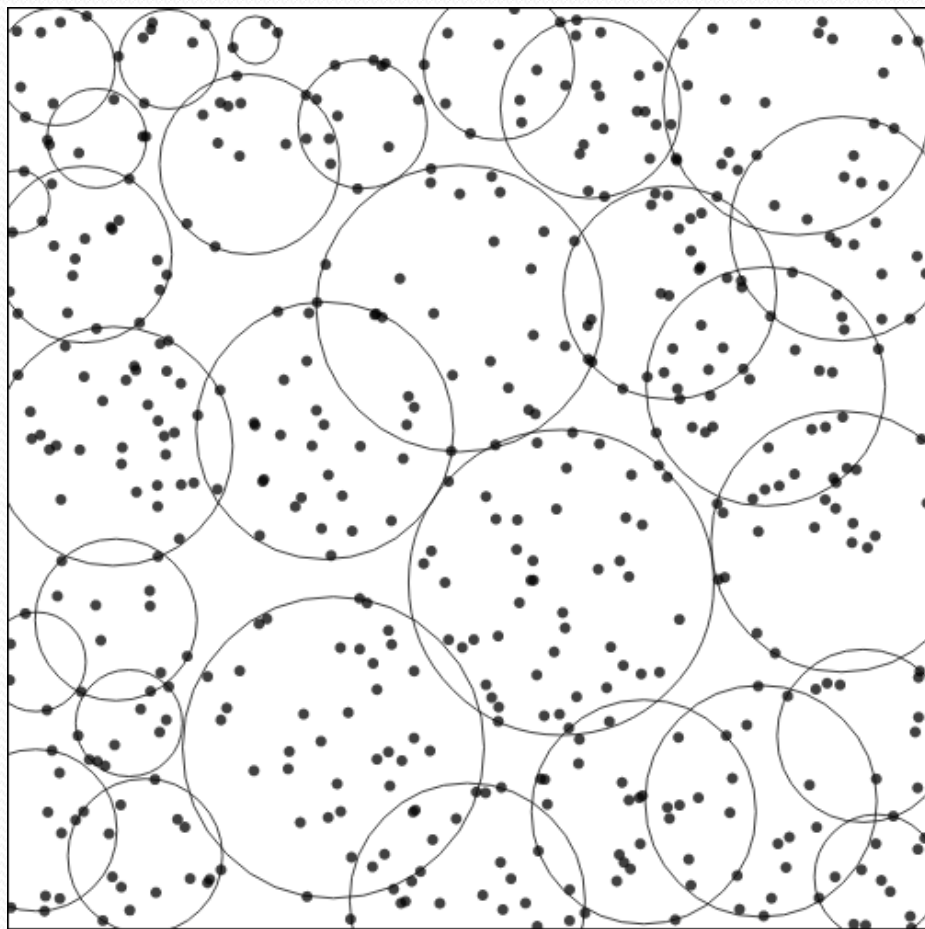


ino4

JPN18

スコア 176846776195

生徒の最良データの紹介

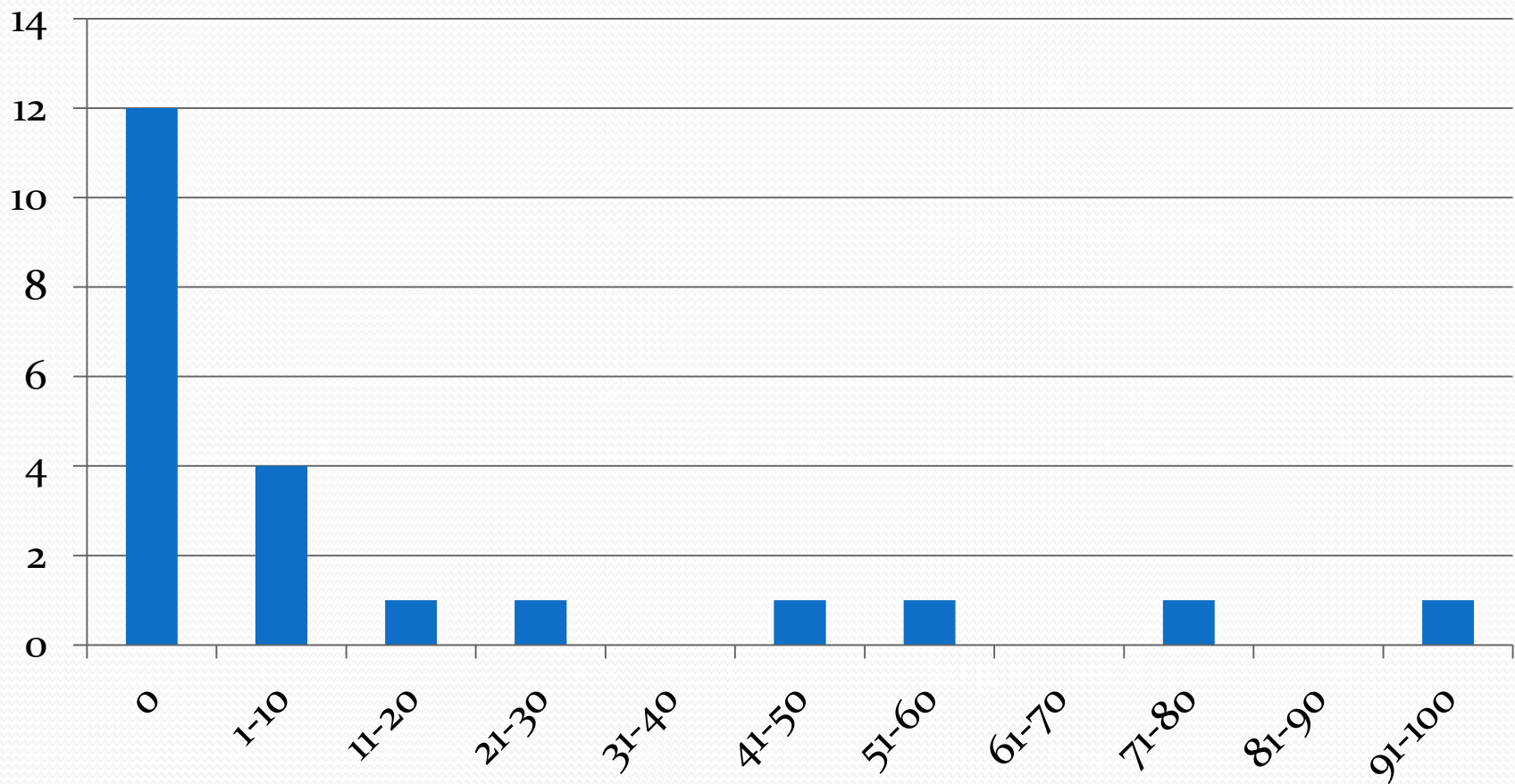


ino5

JPN18

スコア 342198348502

参考：得点分布



まとめ

- 出力のみ問題に特化した戦略を
 - とにかくプログラムを回しまくるとか
- 5時間という短い時間内でできることを
 - この時間内でできる戦略をしよう
- 01→適当な最小包含円+局所探索が良さそう
- 02-05→適当な最小包含円+クラスター分析が良さそう
- まだ改善の余地はあるとはいえ、5時間のコンテストでの成績としては十分ではないでしょうか
 - ただ、不正なデータで0点の人が多いいのは勿体無いですね

ご清聴ありがとうございました

今日もよく寝て万全の体調で3日目に臨みましょう。

ご清聴ありがとうございました

今日もよく寝て万全の体調で3日目に臨みましょう。

(拍手！)