# アルゴリズムとプログラミング実践講座

http://akashi.ci.i.u-tokyo.ac.jp/mary/lectures/algorithm/

火曜　13:00 -- 14:30
I-REF 棟 6階　ヒロビー

稲葉真理　with 浅井大史・手塚宏史

# 先週のトピック

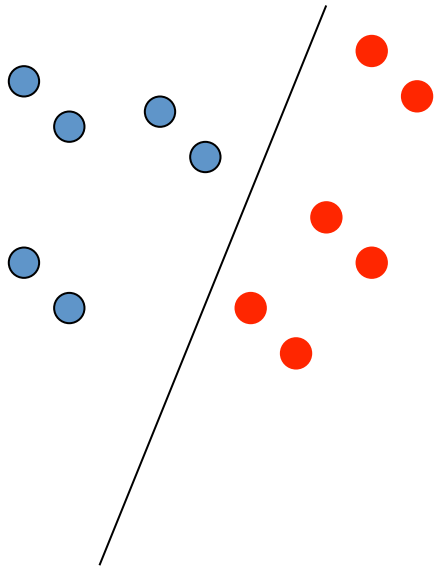
## 計算幾何　と　アルゴリズム

# 座標がある世界・まっすぐな世界

ｎ　点の　２分割は？

では平面上の　ｎ点を直線で2分割だと？

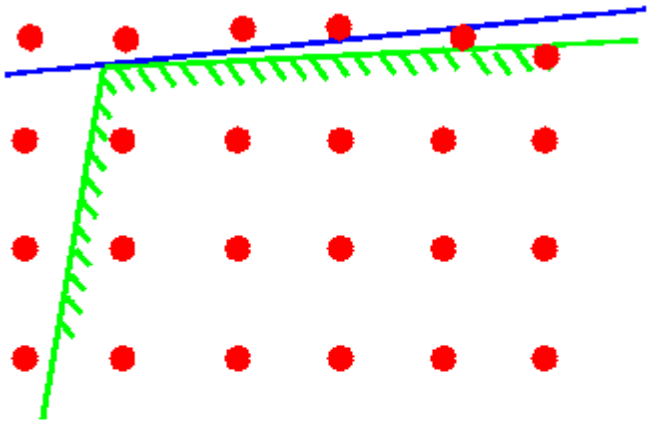
では　ｄ　次元だったら？

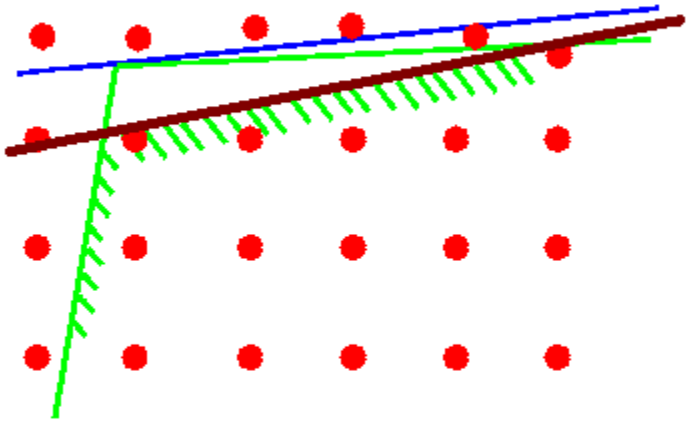


可能性は$O(n^d)$

Linear Separable

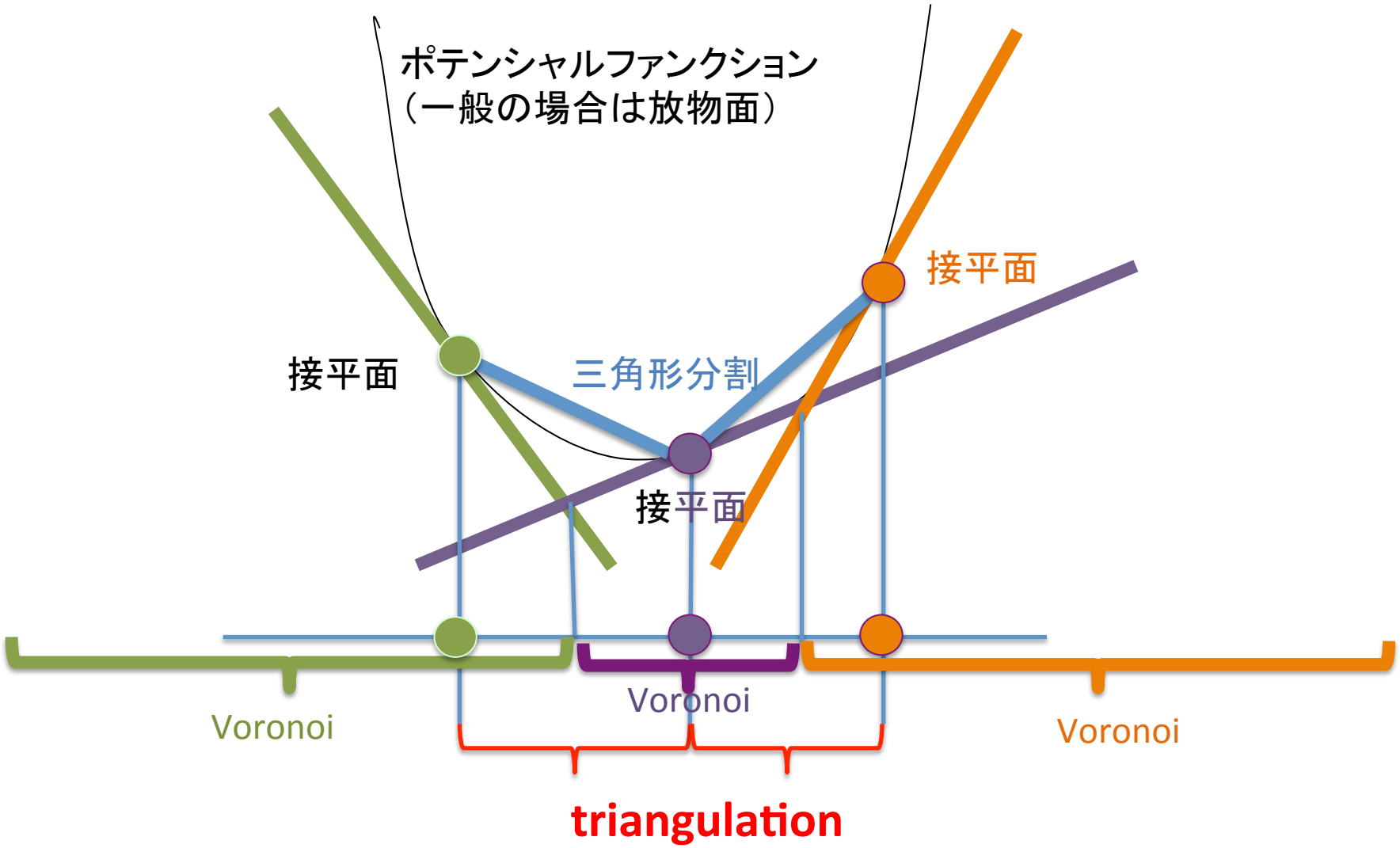
# 整数計画法と分枝切除法





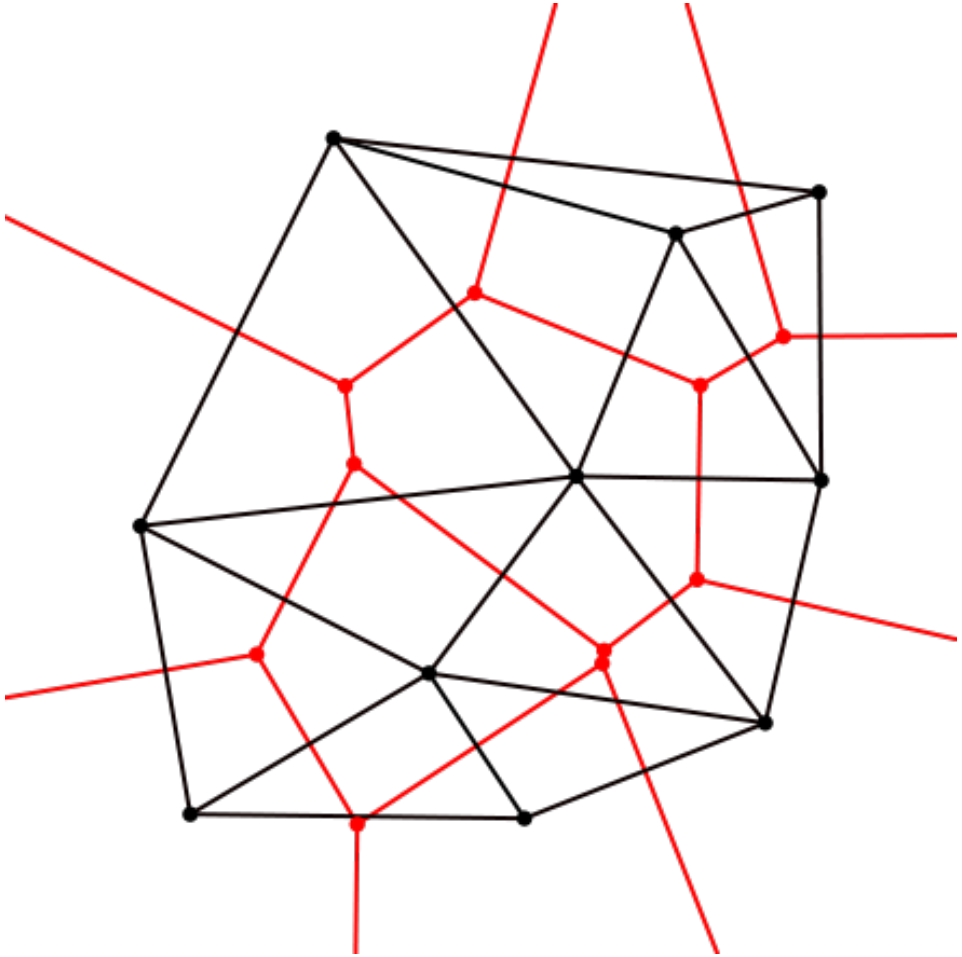
うまく切除平面を加えられると
整数解が切り出せるかもしれない.

# １次元を１次元分あげた接線・接平面

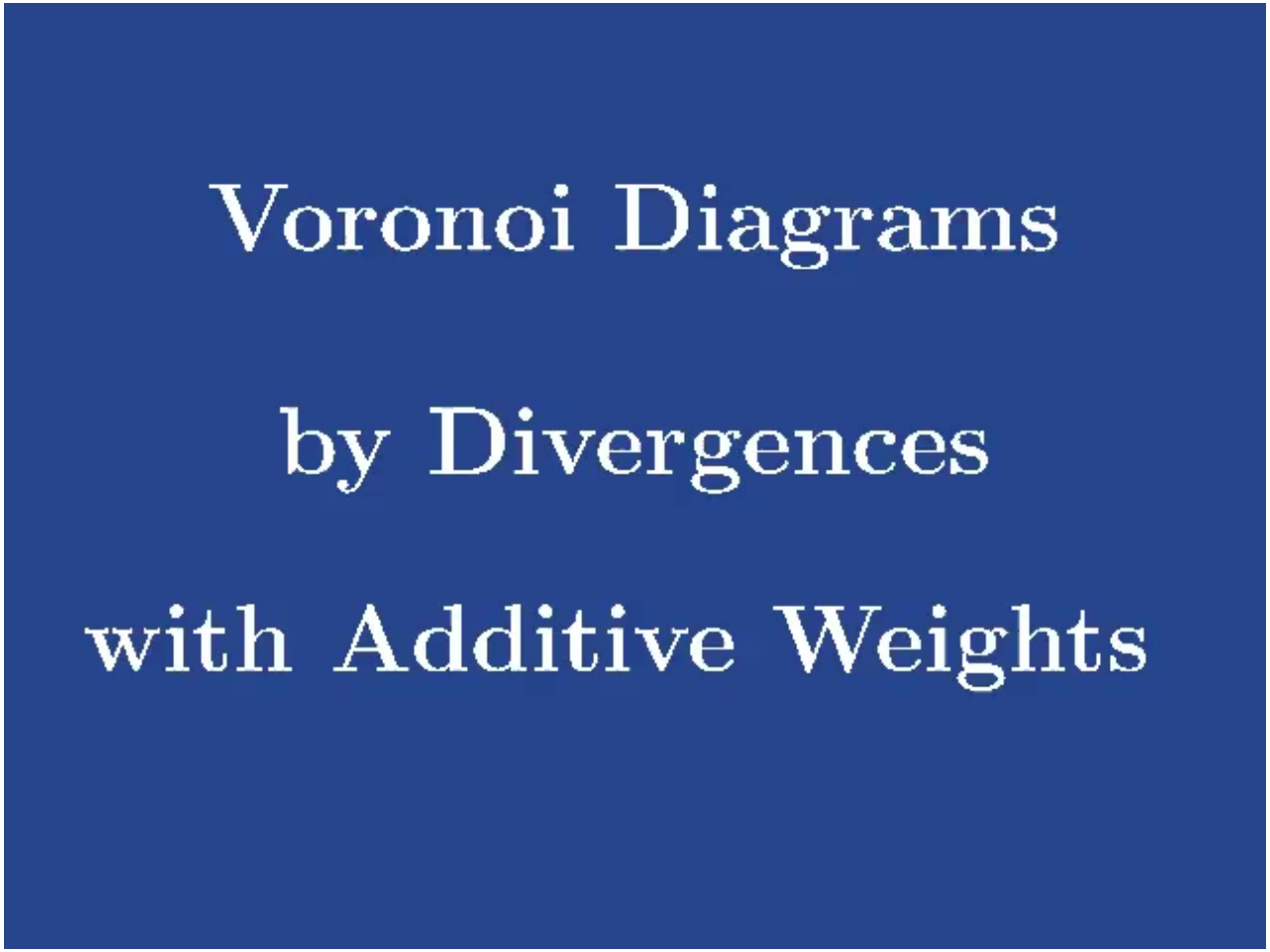


接平面の lower envelope を
プロジェクトダウンしたものが

**Voronoi Diagram**



# Voronoi 図
# [1998年SoCG、音声無し]



# 自分の場合

計算幾何

（論文読む時に、図が多くて素敵♪）

→　クラスタリング

（プリンター　→　color quantization 問題）

→　データマイニング

（クラスタリングの研究中に流行り始めた）

# Color Quantization
# クラスタリングの動画
# [1997年 SoCG、音声無し]

# レポートの〆切について

第三回・第四回は6月22日（日）〆切ます。

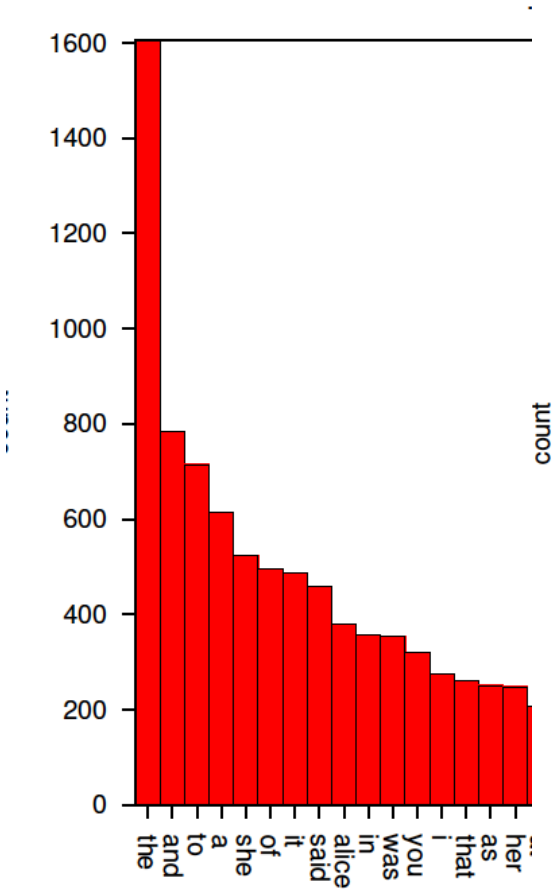
〆切後、いくつかのプログラムを　講義のweb ページで公開する予定です。

# 第三回のレポートから

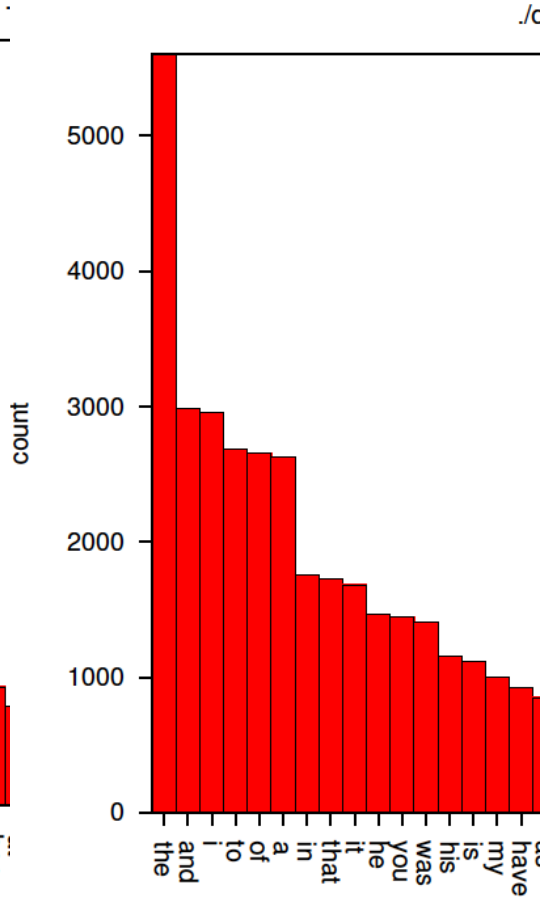
表 2　単語出現数上位 40 種

| rank | Alice's | Sherlock | Scarlet | Baskervilles | MIMI-NASHI | ROKURO-KUBI | YUKI-ONNA |
|---|---|---|---|---|---|---|---|
| 1 | the | the | the | the | the | the | the |
| 2 | and | and | and | and | of | of | and |
| 3 | to | i | of | of | and | and | a |
| 4 | a | to | to | i | to | to | was |
| 5 | she | of | a | to | he | a | to |
| 6 | of | a | i | a | his | he | of |
| 7 | it | in | he | that | in | in | he |
| 8 | said | that | in | it | a | i | her |
| 9 | alice | it | that | he | was | that | she |
| 10 | in | he | was | in | hoichi | you | that |
| 11 | was | you | his | you | that | it | you |
| 12 | you | was | it | was | you | head | as |

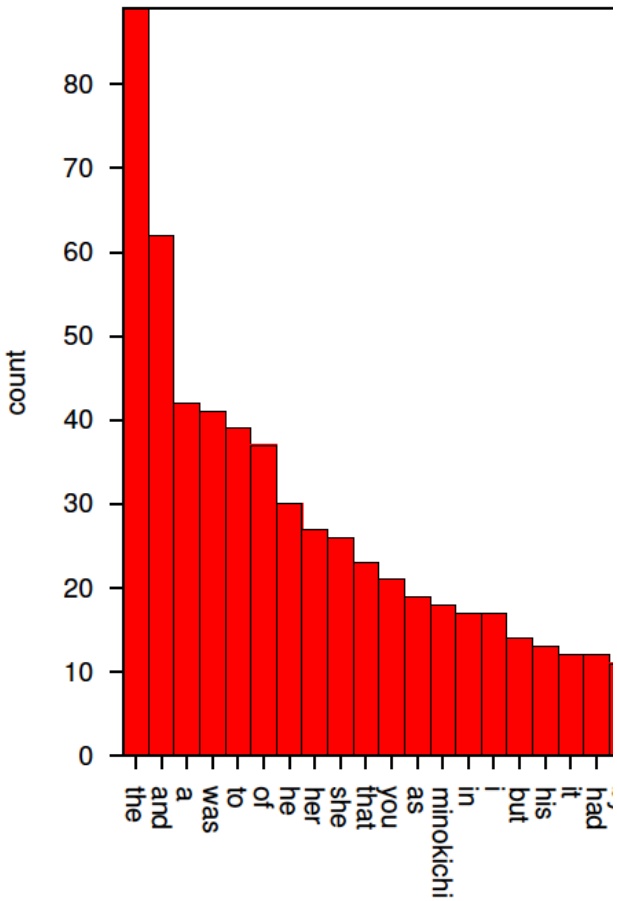
# 第三回のレポートから




Alice




Holmes




Yuki-Onna

# 第三回のレポートから

$$\mathrm{tfidf} = \mathrm{tf} \cdot \mathrm{idf}$$

$$\mathrm{tf}_{\mathrm{i,j}} = \frac{\mathrm{n}_{\mathrm{i,j}}}{\sum_{\mathrm{k}} \mathrm{n}_{\mathrm{k,j}}}$$

$$\mathrm{idf}_{\mathrm{i}} = \log \frac{|D|}{|\{d : d \ni t_i\}|}$$

ここで, TF-IDF は $n_{i,j}$ を単語 i の文書 j における出現回数,

$|D|$ は総ドキュメント数,

$|\{d : d \ni t_i\}|$ は単語　i を含むドキュメント数

# ちょっとだけ補足

bag of words　→　とりあえず文書の順序は無視
(という考え方）　　頻度のみを考えましょう

文書　d
（query するとき使われるであろう）　単語　t

Term Frequency　: $tf_{t,d}$
Document Frequency : $df_t$
Inverse Document Frequency : $idf_t$　　$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}.$

# ちょっとだけ補足
# cf (Collection Frequency) と
# df (Document Frequency)

df は　文書中で何回かは数えない理由の例

| Word | cf | df |
|---|---|---|
| try | 10422 | 8760 |
| insurance | 10440 | 3997 |

► **Figure 6.7** Collection frequency (cf) and document frequency (df) behave differently, as in this example from the Reuters collection.

http://nlp.stanford.edu/IR-book/pdf/06vect.pdf

# ちょっとだけ補足

tf-idf $_{t,d}$:　文書　d内の用語　t　に対する「重み」

1. 少ない文書でしか出現しない単語、その文書では良くでてくる単語は　重い
2. 文書中であまり出てこないとき、そして、沢山の文書に出てくる単語は、軽い。
3. 全ての文書に出てくる単語は最も軽い

# レポート・成績について

- ほぼ毎回、プログラミング課題を出題する予定
  - 効率の良い計算機実験のためのツールを使ってみる
  - アルゴリズムの実装
  - ライブラリの利用・・・　など

- ３回以上、レポートのファイルとプログラムのソースを添付し、メールで提出のこと
  - E-mail: algorithm2014@edu.jar.jp
  - サブジェクト 「アルゴリズムとプログラム実践講座・レポート」
  - 学生証番号と名前は、 メールの本文にも書いてください。
  - プログラムやレポートは（見本として）公開することがあります。適宜、作者名や　コピーライトをいれておいてください。公開不可の場合は、プログラムの冒頭にその旨、コメントをいれておいてください。
  - 質問・作問提案も歓迎 （作問については採用の場合は別途加点）
  - サンプルプログラムは「初心者向け」です。 **上級者は無視してください。**

# 推奨環境など

- Linux, Mac,（Windows+Cygwin）
- 仮想マシン環境（VMware, VirtualBox, Parallels)
  - 余裕があれば、いろいろな組み合わせを試して比較してみると面白いと思います

- 言語
  - 自由。ただし、一般的でない言語については、上記いずれかのOS 上にインストール可能なもの

# 第八回　課題

第七回の課題はお休みです

**第三回・第四回のレポートは**
**6/22（日）で〆切ます。**

# tf-idf の計算

たとえば「プロジェクト・グーテンベルグ」や「青空文庫」から、書籍のデータを、適当な個数収集し、出現する単語について、tf-idf の計算を行い、考察を加えよ。

第三回の課題で tf-idf を計算して、レポートに書いた人は、第八回分も提出済と数えます。

# Clustering　問題

- n個のデータを similarity　に応じてグループわけする問題

# クラスタリング問題

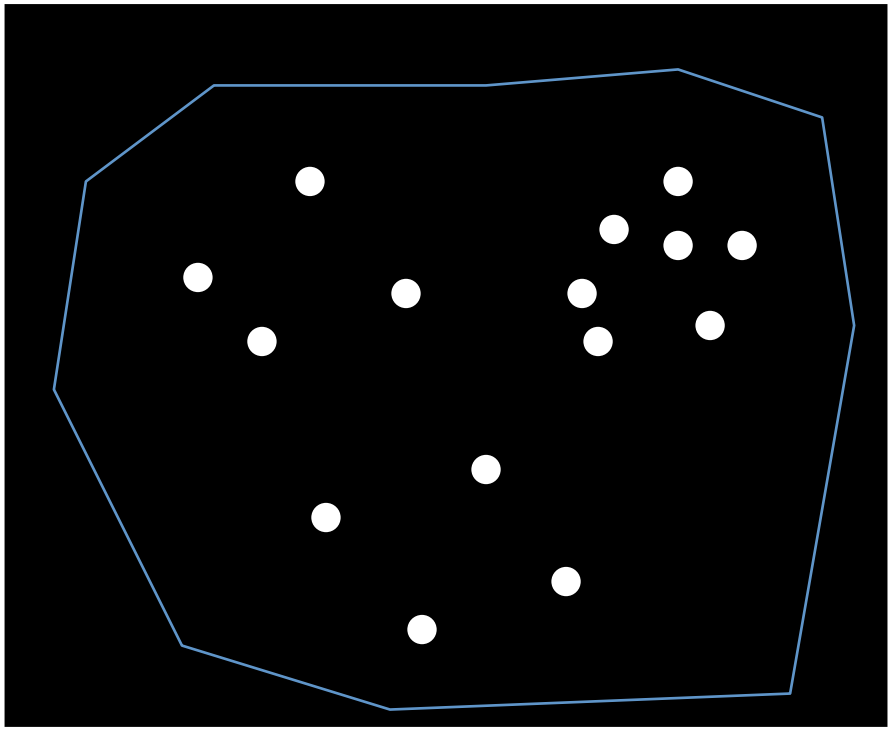
- $n$個のオブジェクトを　$k$個のグループに　分類
  - 類似度（似た物が同じグループに）
  - 評価基準（望ましいグルーピングの指標）

と考えれば、

組み合せ最適化問題

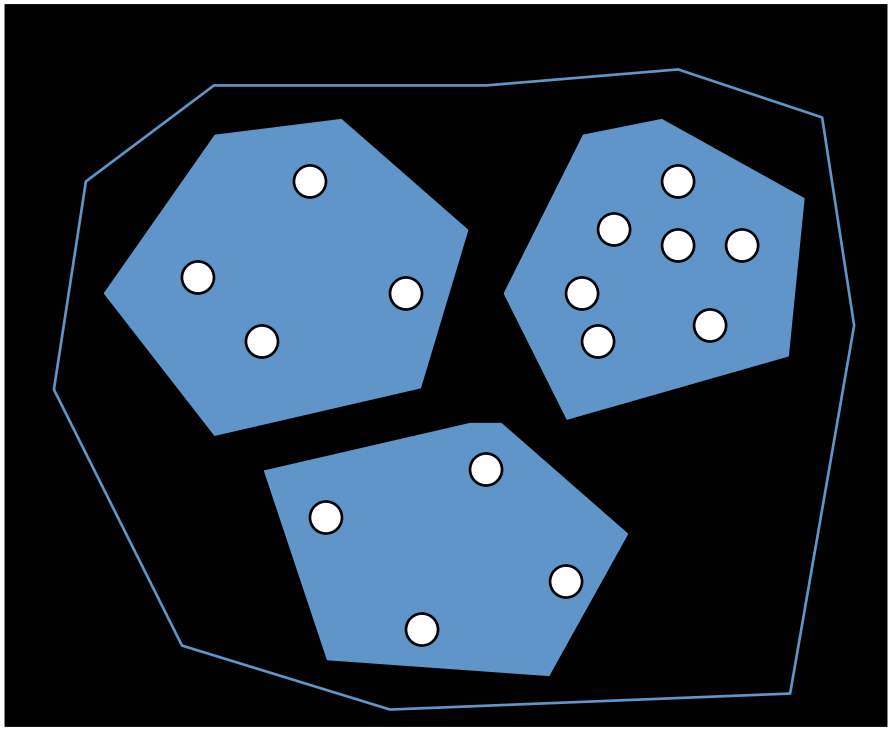
- 一般に NP困難($k$＞3)

# 評価基準



$n=15$

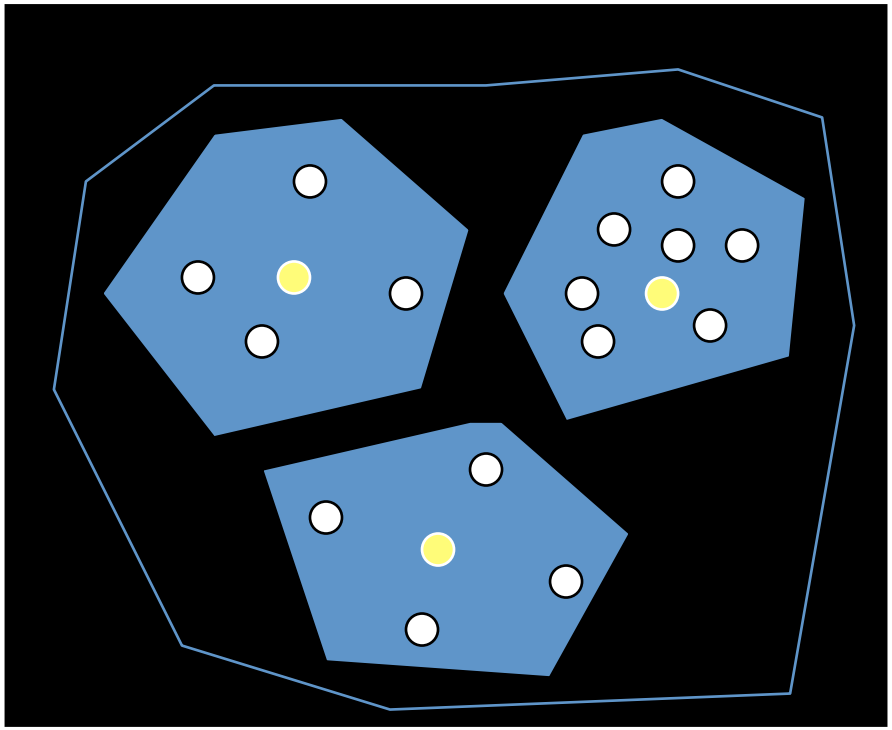
# 評価基準



*n=15, k=3*

# 評価基準

- クラスタ内コスト
- クラスタ間コスト
- 評価基準
  - 最小化



# cluster 問題のcriteria

- Intra Cluster Criteria （クラスタ内コスト）

  たとえば、代表点との距離、全点間距離の sum とか max とか
- Inter Cluster Criteria （クラスタ間コスト）

  全「クラスタ内コスト」の和とか、最大値とか
- 評価関数（目標）
  - minimize

# 無理矢理作った例
# （村の家を、連絡のため、隣組に分割）

- 各家でのろし。うちあげるのろしを小さくすませる
  - min max diameter
- 糸電話を使う。使う糸の総量は短く
  - min sum sum of all pairs distance
- 隣組ごとに連絡所を設置。連絡所はどの家からも近く
  - min max distance from rep
- 連絡所に連絡員を置く。連絡員の歩く距離総和を最小
  - min sum sum of distance
- 全連絡員の中で、一番長く歩く連絡員の距離を最小に
  - min max sum of distance

# 距離行列 vs　幾何的

代表要素の決め方:

CASE 1:　代表は元の要素の一つ

→　整数計画法モデル

CASE 2:　代表はクラスタの要素から計算される

→　（普通は）なんらかの幾何的な特徴が使われている

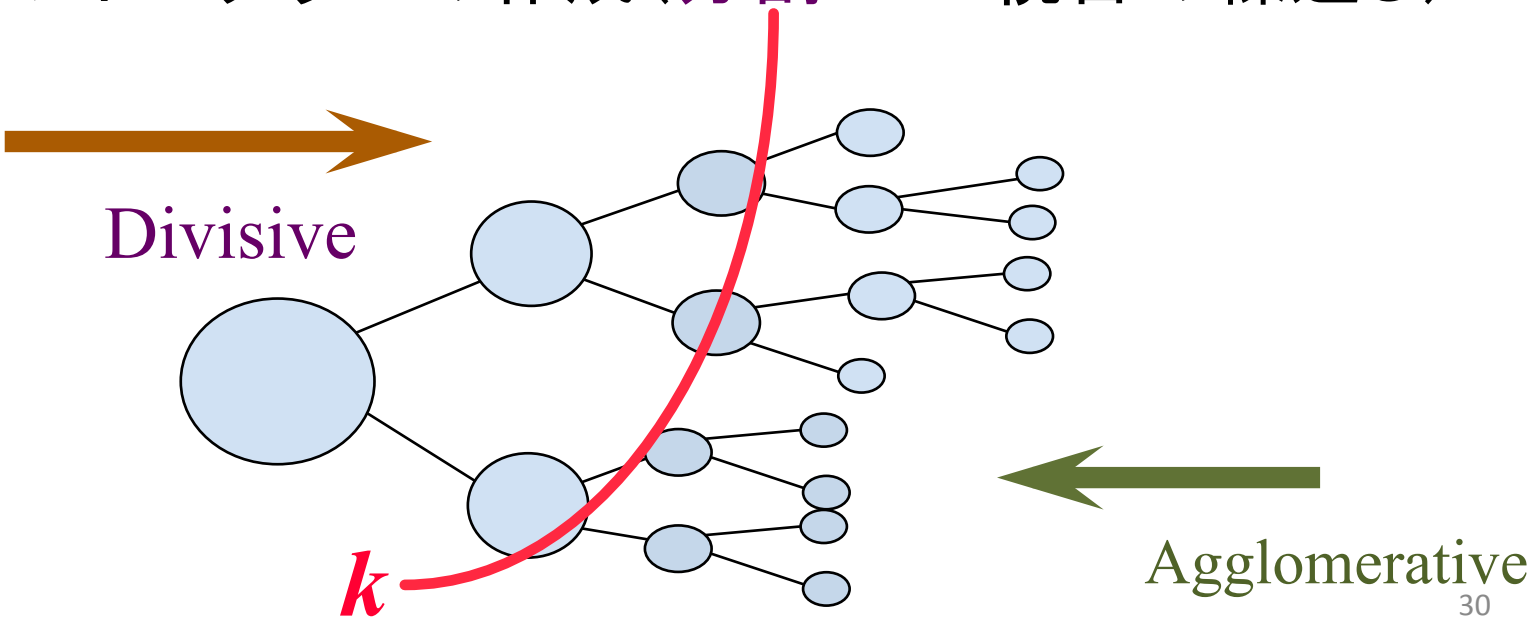
**（注）　各要素どうしの近似度も、いろいろ。**
**matrix で与えられていることもあれば、**
**（i.e., 簡単に計算できない、たとえば統計結果など）**
**計算式(e.x. Euclid 距離）で与えられてることもある**

# 色々な類似度

- ミンコフスキー距離
  - $\{ \Sigma_i |x_i - y_i|^d \}^{1/d}$
  - ユークリッド距離 (d=2)
  - マンハッタン距離(d=1)
- ハミング距離
- ベクトルの角度
- 相関係数
- Kullback–Leibler divergence

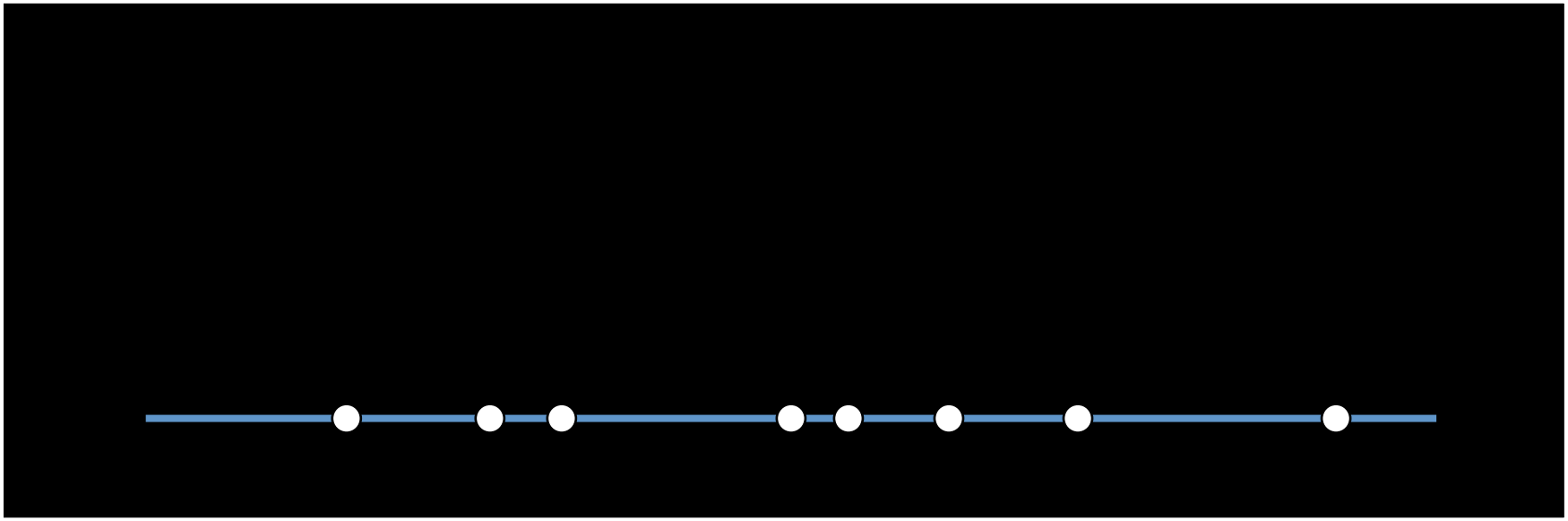
# フラットクラスタリングと階層クラスタリング

- フラットクラスタリング
  - クラスタ間に階層構造がない
- 階層クラスタリング
  - デンドログラムの作成（分割　or　統合の繰返し）



# 加算的幾何クラスタリングの持つ良い構造

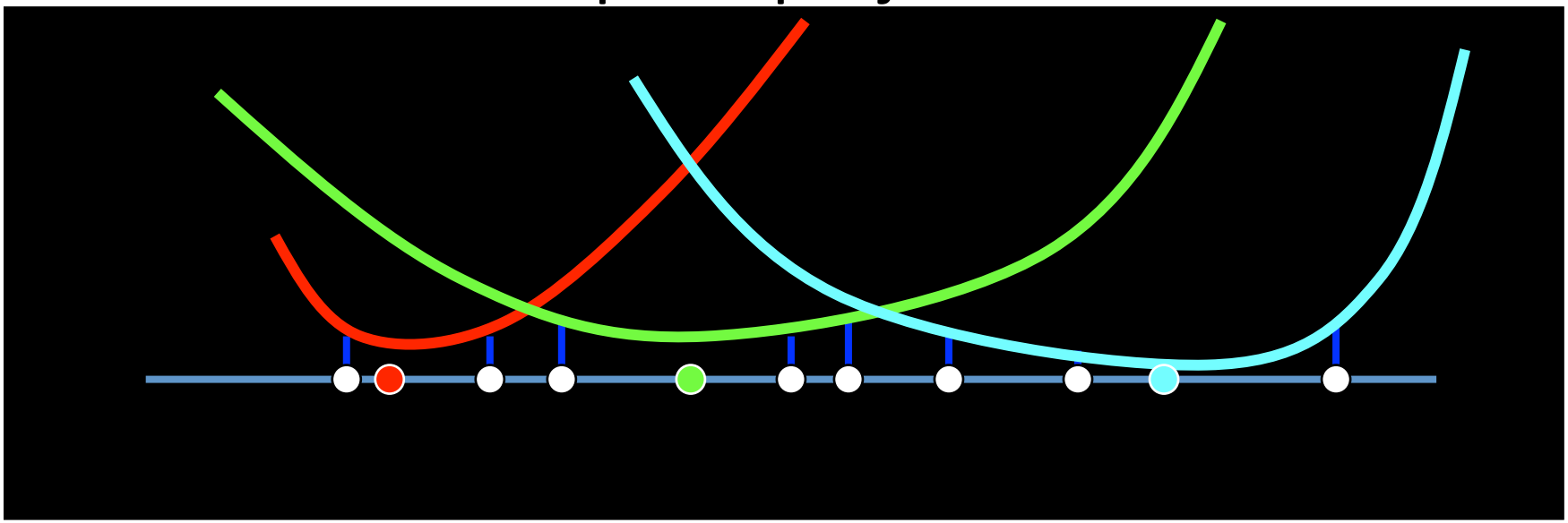
- 空間の分割と考えることができる。
- コスト関数が凸の場合



# 加算的幾何クラスタリングの持つ良い構造

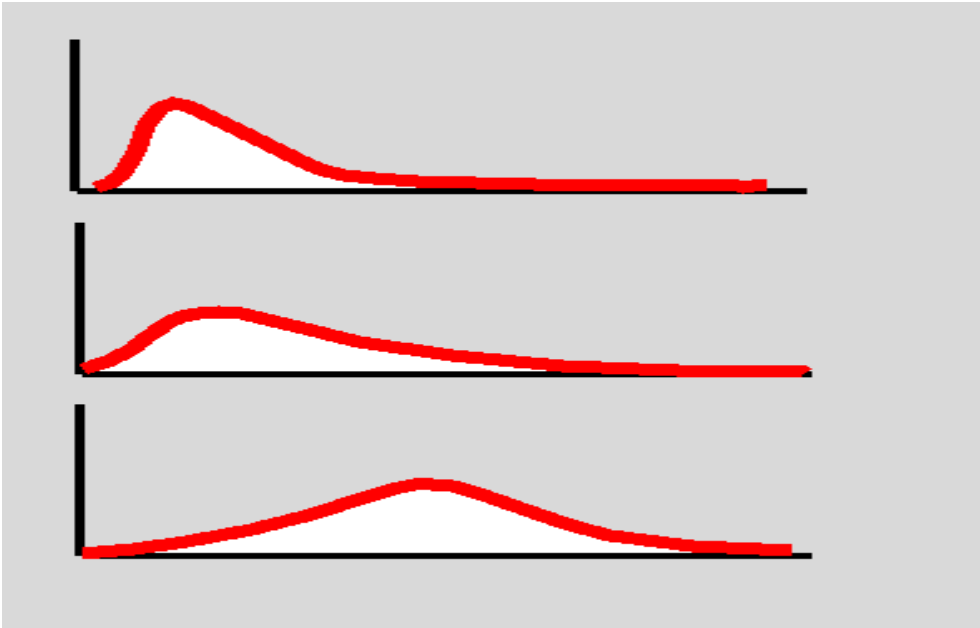
- 空間の分割と考えることができる。
- コスト関数が凸関数の場合
- 別に放物線でなくても構造は同じ

→ lower envelope を project down すれば良い

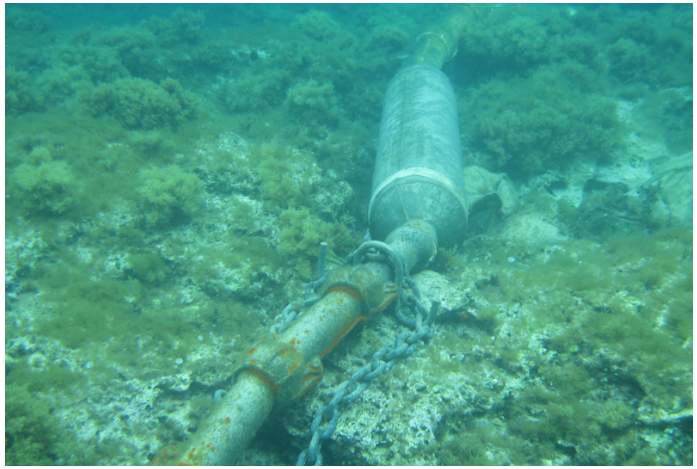


# （おまけ）情報幾何

- 指数型分布族の幾何構造
  - 統計パラメタの張る空間の構造を考える



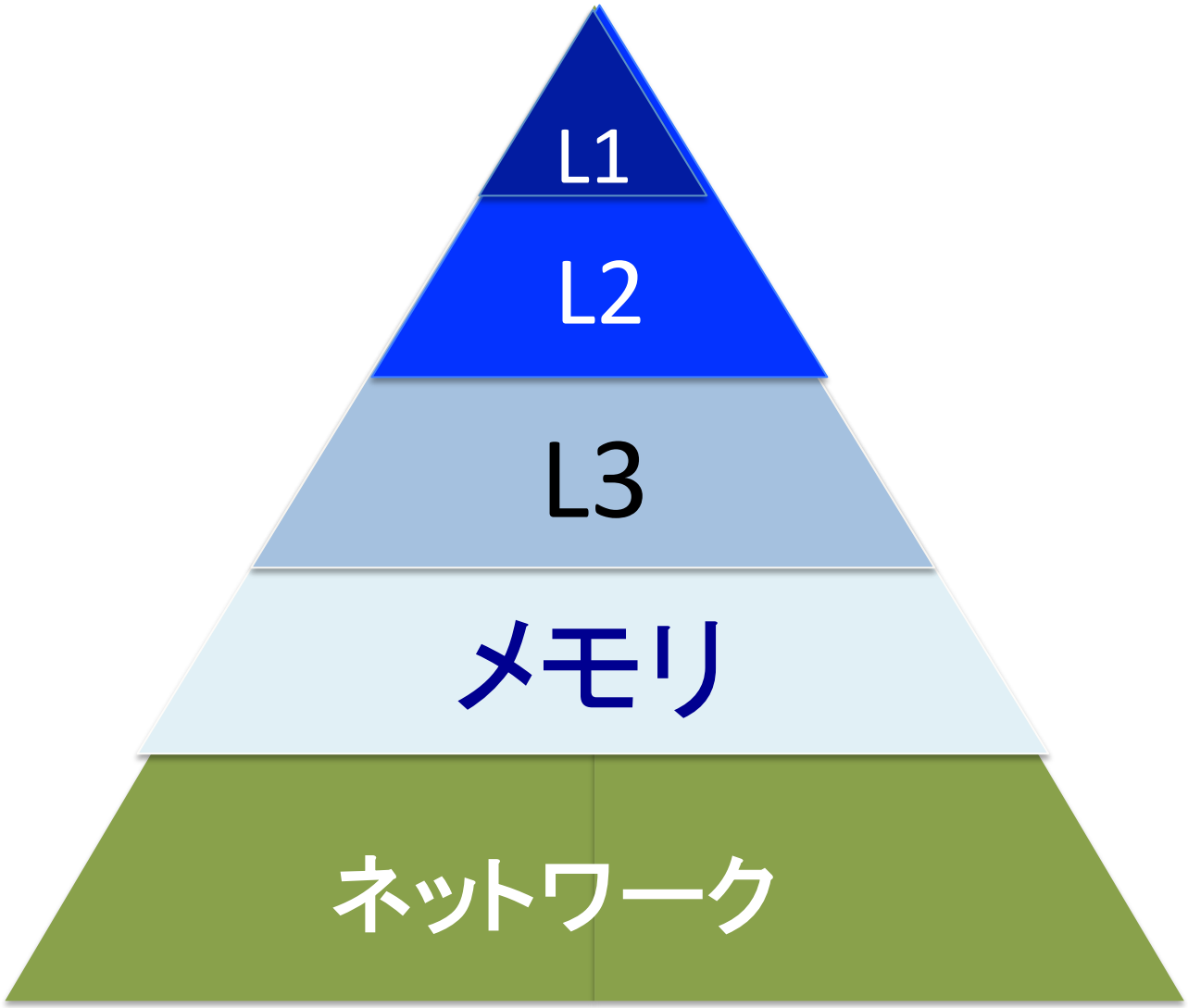
# 先週の伝え忘れ
# バンド幅とレイテンシー



- 線が太くなっても、遅延が減るわけではない。
  （アメリカ東海岸との間では 200msec 越えは普通）
  ファイバーを使っても、光速で通信できるわけではない

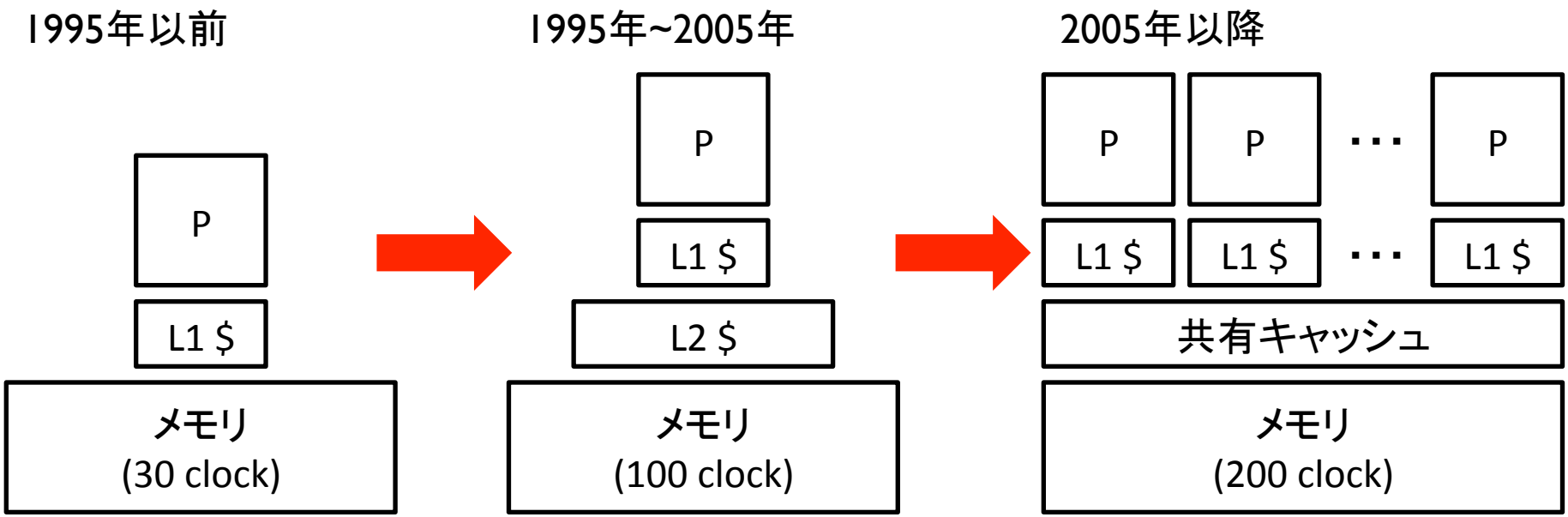
- スイッチでは電気に変換している
  – 光はためておけない

# コアが沢山ある場合は？



# メモリ階層の実現と複雑化

- メモリ階層の利用
  - レジスタファイル
  - キャッシュメモリ (暗黙的なアプローチ)
  - ローカルメモリ (明示的なアプローチ)



# 主として計算科学用
# 並列計算のボトルネック

- 密行列積：演算ボトルネック
- 疎行列：メモリボトルネック
- FFT： バンド幅ボトルネック

# 時代時代ごとのバズワード？

- クラスター
  プロセッサが複数のったボードが
  バスでつながってラックに入っている

- グリッド
  遠くにあるクラスター（あるいはコンピュータ）を
  ネットワークでつなげて使えるようにする。
  （「管理をどうするか研究」に走る　→　終了）

- クラウド
  ユーザに、ハードウェアを意識させるのは面倒。
  「雲のような物なんだよ」と言って、ゆるく提供　←　今、ここ

# 二つのフラットクラスタリング

- Partition　Method

  [Kaufman, Rousseeuw]

  → 代表点はクラスタの要素から選ぶ

- k-means Method

  → 代表点はクラスタの平均を使う

# k-クラスタリングの　整数計画法モデル
## Integer Programming model

dij = d(xi, xj)　xi　と　xj　の距離

yj =　1　　点　xj が representative

　　0　　otherwise

xij=　1　　点　xj によって、点xi が代表される

　　0　　otherwse

$$\min \sum \sum d_{ij} x_{ij}$$

s.t. $\sum y_j = k$ （代表は、全部でk個）

$\sum_{j=1..n} x_{ij} = 1$ (必ずどれかに代表されている）

$x_{ij} \leqq y_j$　　（矛盾しないために）

# Partition　Method

## PAM　（k-medoid -- sum of L1)

### BUILD（incremental)

まだ選ばれていない i, j ( i がcandidate)

d(i,j) と　Dj　を比較　　Djは、今までjの値

Cji= max (Dj　ー　d(i,j), 0)

$Ti = \sum_{j} Cji$

maximize　Ti (Ti　が最大な　i を探す）

### SWAP　（i と　h　の交換を考える）

# Partition　Method

PAM　（k-medoid -- sum of L1)

BUILD（incremental)

SWAP　（i と　h　の交換を考える）

Cjih =　0　：　jに近い　representative は

i でも　h でもない

Cjih = d(j, h) – d(j,i) : １番がi、　２番が　h

Cjih = Ej – d(j,i)　：　Eij　２番目に　j　に近い距離

Tih = ∑　Cjih　(j について足す）

minimize　Tih

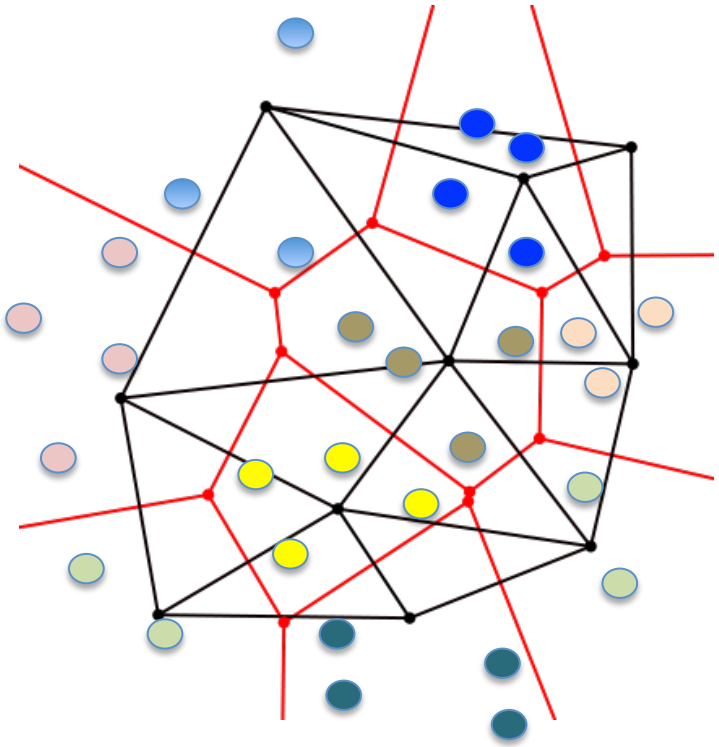
# PAMの性質

1. せいぜい　５０個が限界（論文発表当時）
2. データの与え方の順番に関係しない

Random Sampling して、扱えるData　を増やす。

# 距離和の和を最小にする
# クラスタリング

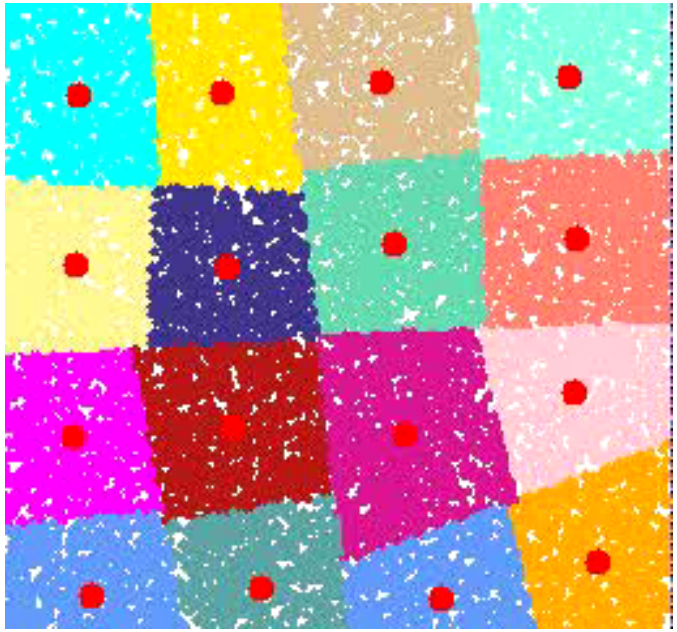
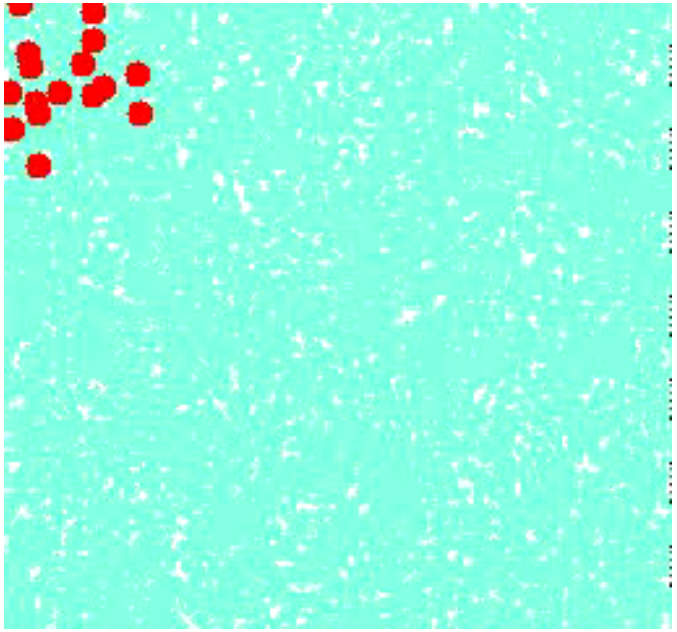
→　その距離での
Voronoi 分割になる



# k-means法

1. 適当にクラスタを作る

2. 目的関数を計算
3. クラスタの重心を求める
4. 重心に近い要素で、新しくクラスタを構成

5. 構成がかわったら2へ。 変化なければ終了

# k-means　の初期値による違い[動画]
# （最適解にはたどりつかない）