

「第1回量子コンピュータ勉強会」
～量子コンピュータってなんだろう？まずは触ってみよう～

日時：12/13(月) 20:00-21:00

場所：オンライン

四谷ラボ

山本 慎也

目的

このセッションが終わるころには・・・

- 量子アニーリングのプログラミングっ想像つかなかったけど、なんとなく理解できた気がする。
- 知り合いに「量子コンピュータ触ったことある」と自慢できる。
- 世の中の組合せ最適化問題を探したくなる。
- 量子アニーリングをビジネスで利用したい。

座学編



$$R_1 = \frac{35}{8}$$

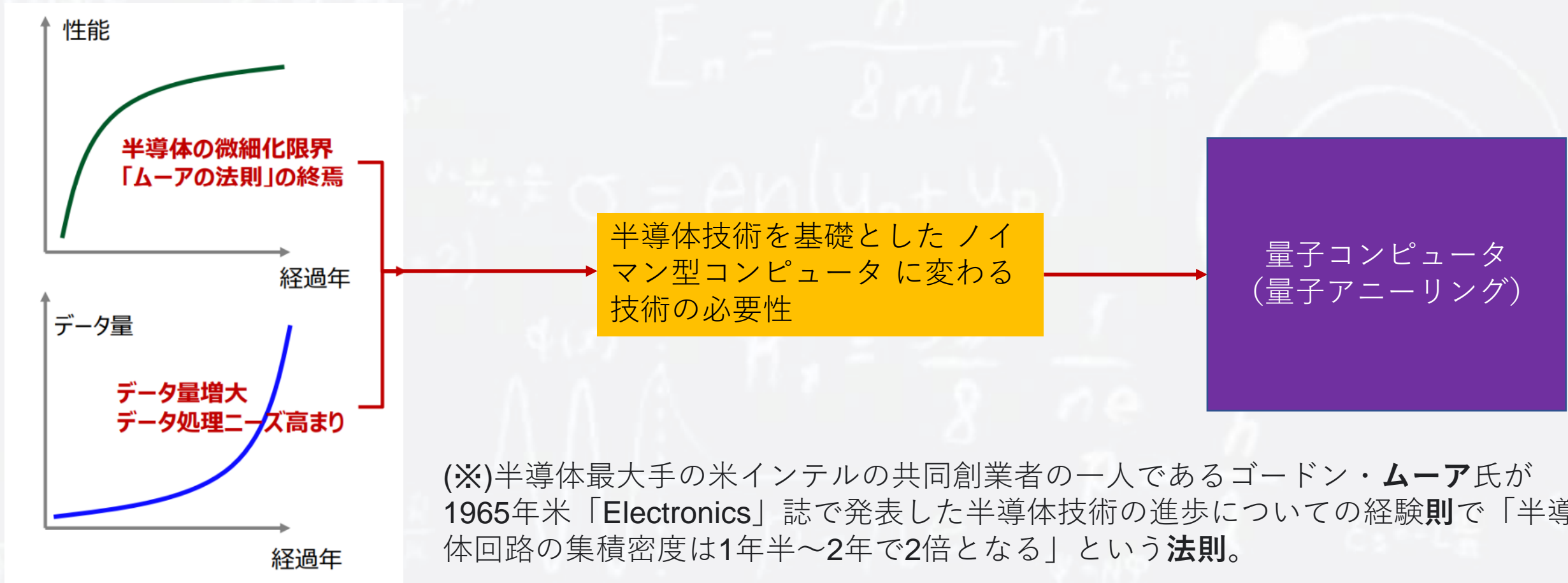
$$\frac{r}{ne} = \frac{h}{p}$$

$$p = p_0 e$$

$$E_2 = -L \frac{11}{n}$$

まずは、量子コンピュータの必要性

○「ムーアの法則(※)」の終焉、データ処理のニーズがより一層高まっている事などを受けて、新たな原理を用いた次世代型コンピュータへの期待が高まり、世界中で量子コンピュータの開発が加速。



量子コンピュータへの期待

- Googleは、量子論理ゲートで構成される「Sycamore」という名前の新しい54量子ビットのプロセッサを開発
- スパコンでは1万年かかる「ランダム量子回路サンプリング」と呼ばれる計算タスクを200秒で計算したと発表

Google量子超越性 1万年 → 200秒

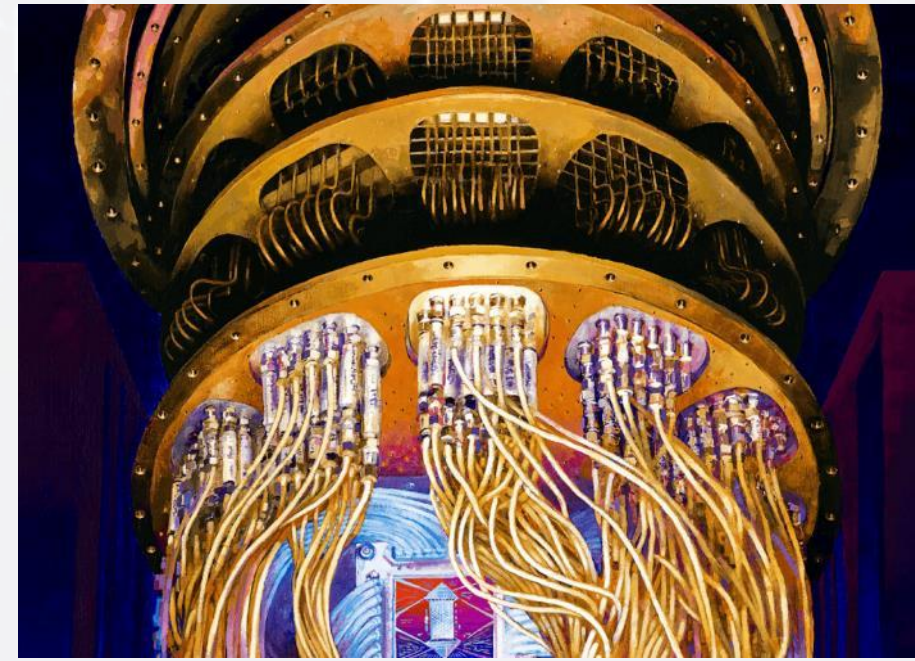
量子コンピュータとスパコンが、速さを競うドッグレースみたいなもの。スパコンが勝てば今の古典コンピュータはしばらく安泰。量子が勝てば「量子超越（Quantum Supremacy）」となって、今のパソコンもうかうかしてられない、ということになります。古典コンピューターの技術者は、うかうかしてられないという結果に！



👉前アメリカ大統領の娘のイヴァンカさんが「アメリカはあ、Quantum Supremacy達成しまあしたー。」とツイート。

量子超越実験に使用したであろう量子コンピュータ（知らんけど👉

機能を追求すると結果、美しくなる。



量子コンピュータとスパコンの違い

| スーパーコンピュータとの主な違い | | | スーパーコンピュータの主な用途 | | |
|------------------|--|--|--|--|------------------------------------|
| | スーパーコンピュータ | 量子コンピュータ | <ul style="list-style-type: none"> 主に、気象予測、災害予測や創薬、構造解析など再現困難なコンピュータシミュレーションに活用 基礎研究からものづくりまで、あらゆる研究開発において必須のツールとなっている | | |
| 目的 | 高度な数値計算やデータ処理のすべて | 限られた目的に大きな力を発揮 | | | |
| 演算単位 | 0 or 1 (ビット) | 0 and 1 (量子ビット) | | | |
| 計算方法 | すべての入力に対して 毎回計算 | 重ねあわせ状態を利用して一括計算 | | | |
| 強み | <ul style="list-style-type: none"> 既存技術を活用 汎用性の高さ | <ul style="list-style-type: none"> 特定のタスク、問題を高速に処理できる 電力消費量が少ない (超電導技術を使う場合) | 分野 | 概要 | 国内取組例 |
| 弱み | <ul style="list-style-type: none"> 消費電力が大きい 微細加工技術の限界 入力数が増えると、計算コストが増加 | <ul style="list-style-type: none"> 製造や制御が難しい (量子状態の維持が困難) | 気象予測・災害予測 | 防災、減災に資する地球変動予測 | 海洋研究開発機構 |
| 見通し | <ul style="list-style-type: none"> 今後も使い続けられるが、性能向上の度合いは、鈍化する可能性有り | <ul style="list-style-type: none"> 限られた問題に対して実用化が進展、高速化が期待されている | 医療・創薬 | 創薬、薬の人体への作用予測 | 理化学研究所 |
| | | | 分子シミュレーション | 新物質、エネルギーの創成 | 東京大学物性研究所、分子科学研究所 |
| | | | ものづくり | 革新技術の創出等、次世代ものづくりへの活用 (流体制御、大型プラントの耐震設計等のシミュレーション) | 東京大学生産技術研究所、日本原子力研究開発機構、宇宙航空研究開発機構 |
| | | | 構造解析 | 物質と宇宙の起源と構造解析 | 筑波大学計算科学研究センター、高エネルギー加速器研究機構 |

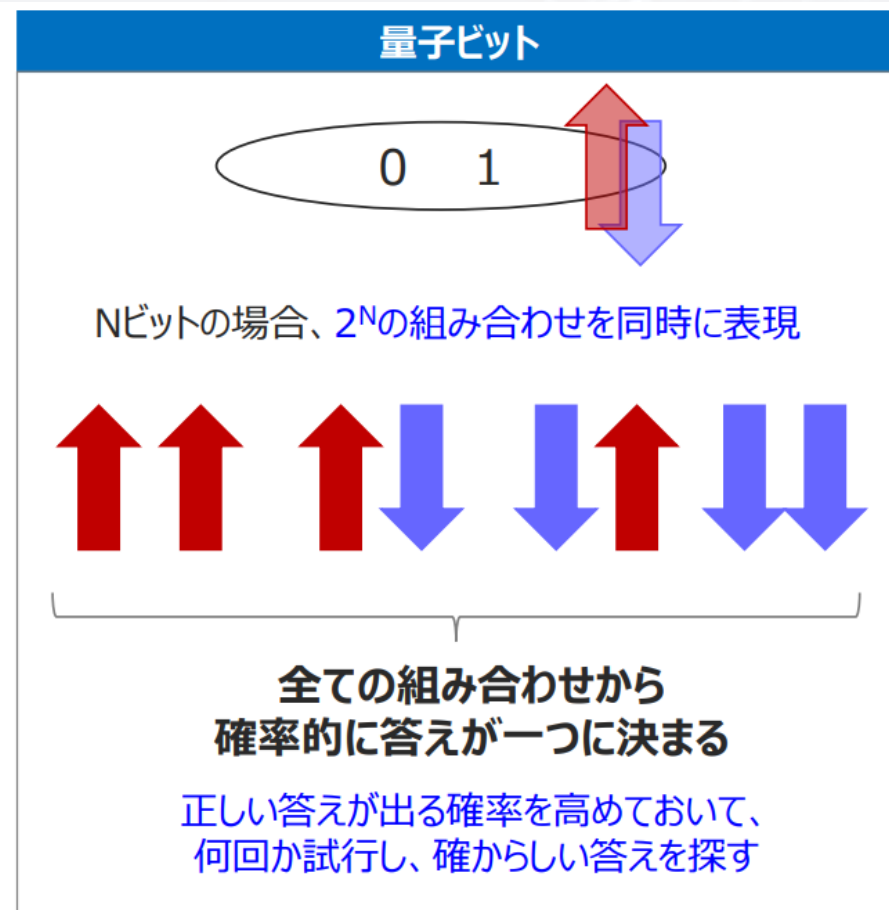
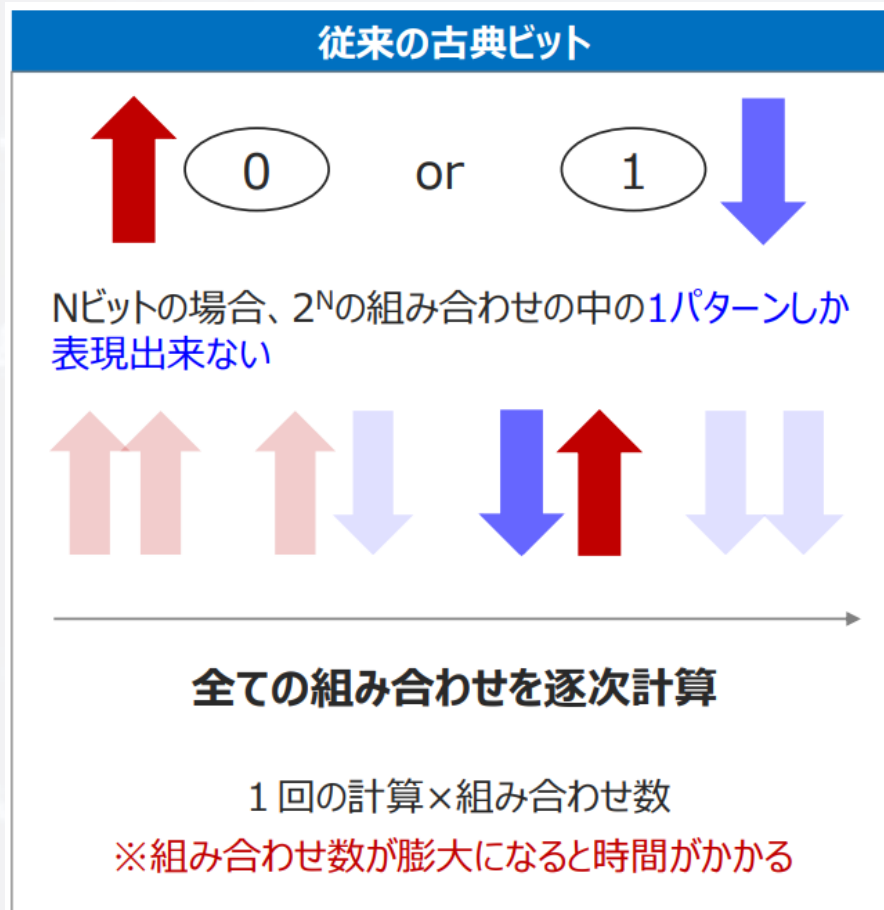
参考：東工大理学院 物理学系の西森秀稔教授による量子コンピュータに関するプレスセミナー資料を基に加筆修正 (2017/12/12)

参考：理化学研究所 計算科学研究センターHPを基に作成
(<https://www.r-ccs.riken.jp/jp/fugaku/target>)

Copyright (c) 2020 The Japan Research Institute, Limited

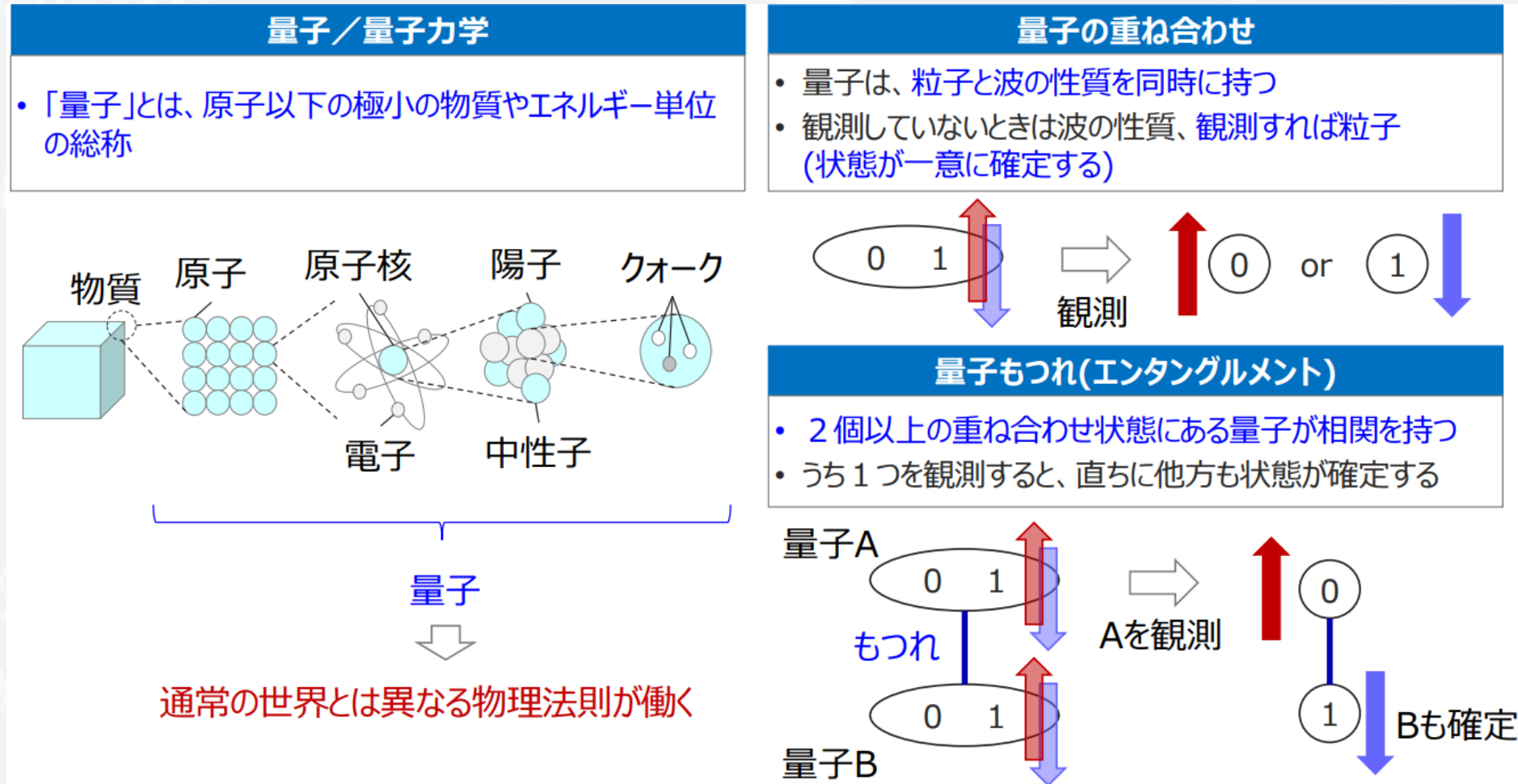
量子コンピュータを量子ビット

- 量子コンピュータとは、**量子の不思議な振る舞い**を情報処理に応用したコンピュータ
- 従来の古典ビット 0 か 1 かでなく、0 と 1 の「**重ね合わせ状態**」という量子も持つ**量子ビット (Qbit)** により実現される



量子ビットを支える量子の振る舞い

○ 量子が備える「重ね合わせ」、「量子もつれ」(※)などの特性を利用して実現



(※) 重ね合わせ、量子もつれ <https://www.ibm.com/blogs/think/jp-ja/ibmq-principle-of-quantum-computer/>

量子コンピュータの種類

| | 汎用型 | 特化型 |
|-------------|---|---|
| 解ける問題 | 汎用 ※量子アルゴリズムが確立されている 問題のみ高速化可能 | 組み合わせ最適化問題 |
| 業務適用 | 実用化まで 10年以上 | 実用化に向けた 取組み多数 |
| 規模 (商用化) | 53量子ビット (IBM)  出所：IBM社 公式HP Blog「HP量子優位性時代の到来の鍵となる量子ボリュームとは？」 (https://www.ibm.com/blogs/think/jp-ja/power-quantum-device/) | 2,048量子ビット (D-Wave)  出所：D-Wave社 公式HP (http://dwavejapan.com/system/) |

量子コンピュータの種類（詳細）

| | 量子コンピュータ | | | 古典コンピュータ |
|-------------|---|------------------------------|------------------------------|-----------------------------|
| 方式 | 【汎用型】 量子ゲート方式 | 【特化型】イジングモデル方式 | | |
| | | 量子アニーリング方式 | レーザー ネットワーク方式 | シミュレーション方式 (量子インスパイアド方式) |
| 適用 領域 | 汎用的な 計算を実現 | 組み合わせ最適化問題に特化 | | |
| 環境 | 極低温／超高真空(*) | | 常温／常圧 | 常温／常圧 |
| ハード | 超電導量子回路(*) | | 光パラメトリック発振器 | 従来の半導体 |
| 主な 推進組織 | IBM/Google/ Microsoft/Intel/ Alibaba/Q-Leap | D-Wave/QEC/ NEC/NEDO | NTT/NII (国立情報学研究所) /量子人工脳 | 富士通/日立/ 東芝/NEC |
| 規模 (商用化) | 53ビット (IBM) | 2,048ビット (D-Wave Systems) | 2,000ビット (NTT&NII共同開発) | 8,192ビット以上 |

(*)主流の方式

量子アニーリングの誕生

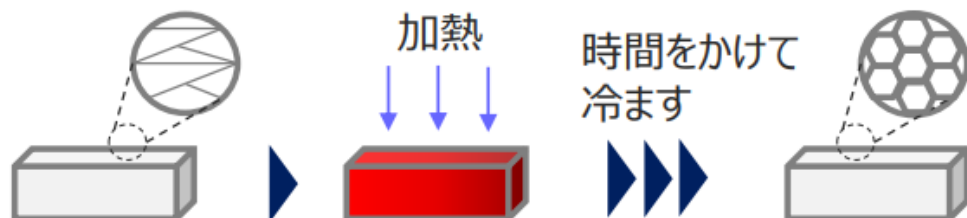
- 日本生まれ（カナダ育ち？）
- 1998年に東京工業大学の門脇氏と西森氏が提案
- 2011年に量子アニーリング商用ハードウェアD-Waveが発表
- 「**組合せ最適化問題**」 **特化型**の量子コンピュータとして誕生

量子アニーリングの動作イメージ

由来／処理イメージ

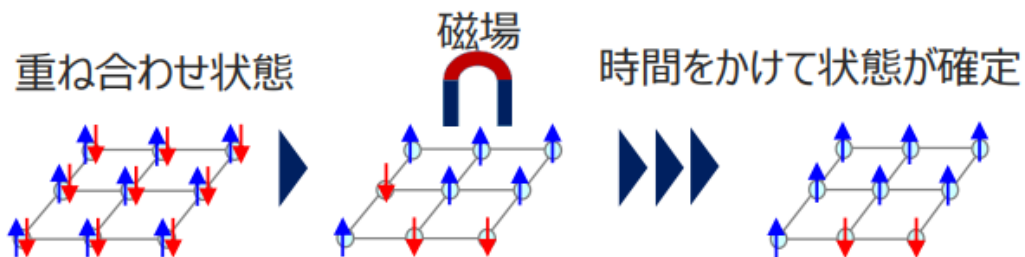
焼きなましのイメージ

金属中の欠陥が消滅し、ひずみのない等方的な結晶になる

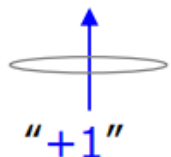


量子アニーリングのイメージ

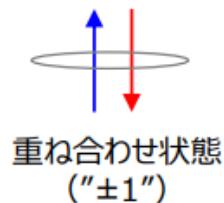
各スピンの状態が安定していき、各スピンの向きが確定する



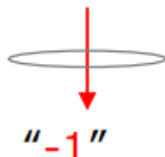
※スピン(=量子ビット)



"+1"



重ね合わせ状態
("±1")

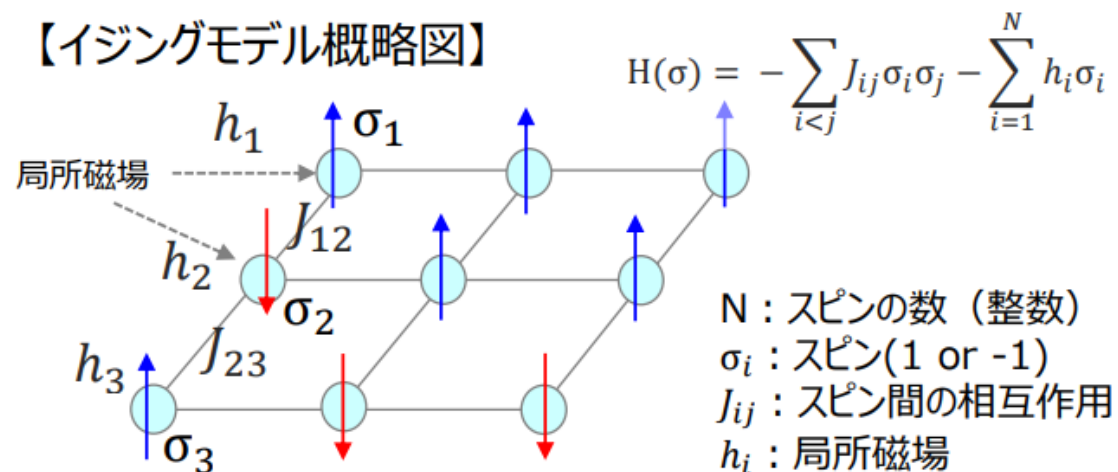


"-1"

イジングモデル

- 「イジングモデル」は、上向き、または、下向きのスピンから構成され、隣接するスピン間の相互作用および外部から与えられた磁場の力によって状態が変化
- 最終的に、ハミルトニアン(系全体のエネルギーのこと)が最小の状態でスピンは収束

【イジングモデル概略図】



【QUBO(Quadratic unconstrained binary optimization)】

・イジングモデルは{1,-1}で表現するが、QUBO $H(q)$ は{0,1}で表現

$$H(q) = -\sum_{i < j} Q_{ij} q_i q_j$$

q_i : {0, 1}を取る変数
 $q_i = \frac{\sigma_i + 1}{2}$ の関係

量子アニーリングの処理の流れ

① 初期化

量子ビットに磁場（電磁波）を加えて、全ての解候補を表現した重ね合わせ状態にします。

この時、量子ビット間の相互作用は弱く、各量子ビットの状態も0か1の定まらない状態です。相互作用とは、量子ビット間で互いの状態に対して影響を及ぼし合う力で、数式における係数に相当します。

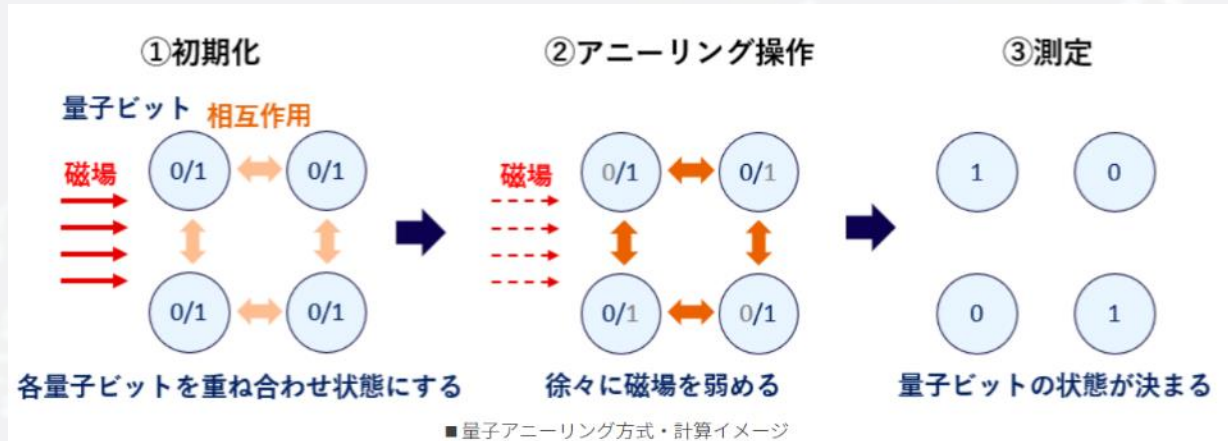
② 量子アニーリング操作（計算試行）

磁場を徐々に弱めるのと同時に量子ビット間の相互作用を強める操作を行うことで、各量子ビットは重ね合わせ状態から0か1に決定された状態に変化します。

量子ビットの状態は、**量子ビット間の相互作用により数式の値が小さくなるような状態に変化**していきます。

③ 測定

最終的には、量子ビットは0か1に確定した状態になるので、その状態を測定します。



このように量子アニーリング方式では、全ての解の候補を同時に表現し、そこから解を絞り込むことで高速計算の実現を目指しています。

組合せ最適化問題とは

組合せ最適化問題とは、様々な制約の下で多くの選択肢の中から、ある指標（価値）を最も良くする変数の値（組合せ）を求めることです。代表的な組合せ最適化問題に巡回セールスマン問題などがある。

次ページで次の組合せ最適化問題を紹介

- 菓子選択問題
- 長方形詰込み問題

菓子選択問題（組合せ最適化問題例）

○学校遠足の菓子選びを思い出してみましょう。この場合、変数・制約・価値は下記で与えられます。

変数：どの菓子を買うのか？（ x_1 の値、1:買う、0:買わない）

制約：遠足で決められた総額 300円

指標：満足度

| お菓子 | 変数 | 値段 | 満足度 |
|----------|-------|------|-----|
| ポッキー | x_1 | 120円 | 10点 |
| チュッパチャプス | x_2 | 30円 | 3点 |
| カントリーマアム | x_3 | 90円 | 4点 |
| ポテチ | x_4 | 150円 | 8点 |
| 果汁グミ | x_5 | 120円 | 7点 |

指標： $10 x_1 + 3 x_2 + 4 x_3 + 8 x_4 + 7 x_5$

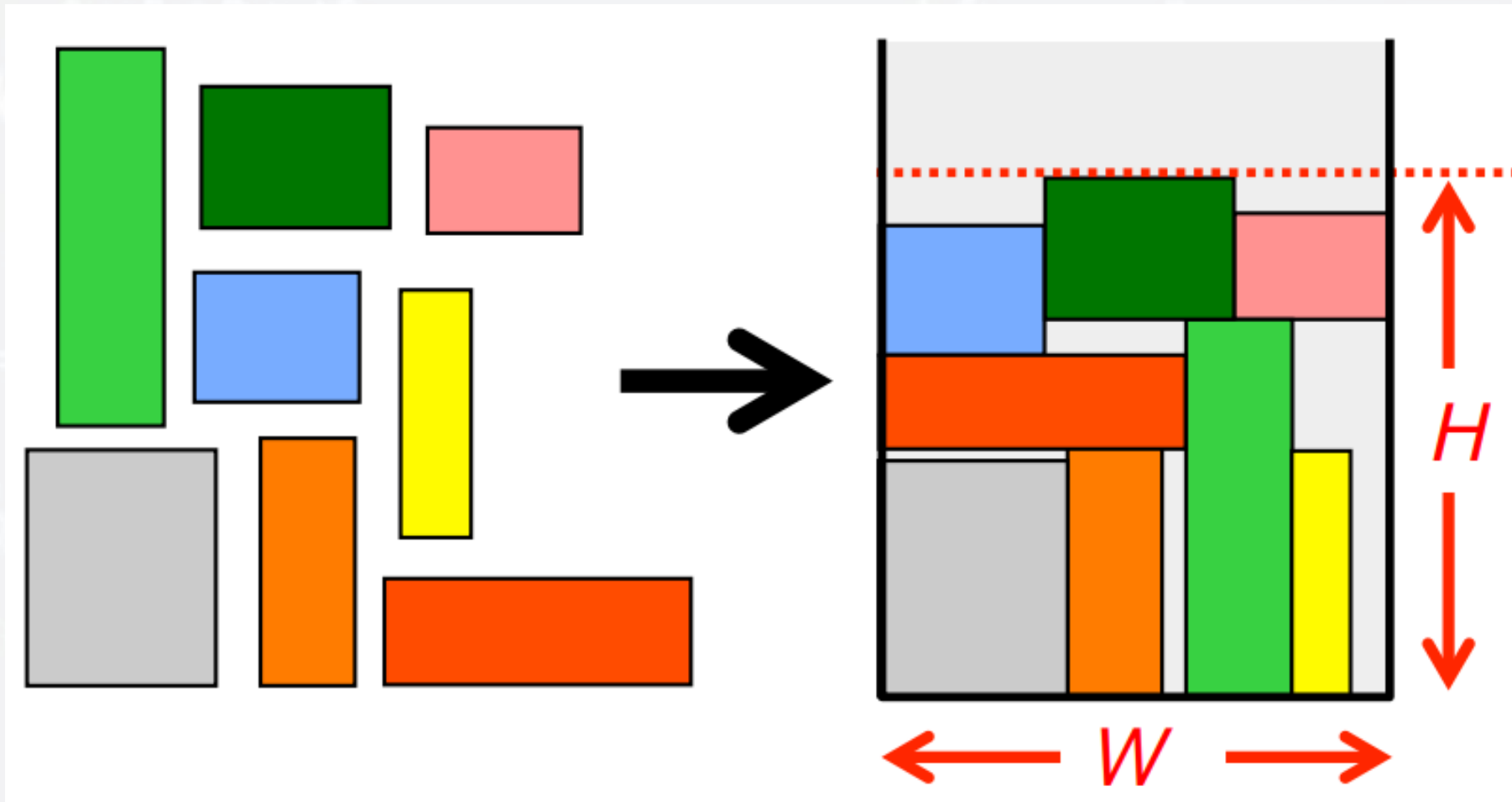
制約： $120 x_1 + 30 x_2 + 90 x_3 + 150 x_4 + 120 x_5 \leq 300$

この**制約**を満足しつつ、**指標**を最大化する菓子の組合せが求めたい答えとなります。

答え：ポッキー、チュッパチャプス、ポテチ（満足度21点、総額300円）

長方形詰込み問題（組合せ最適化問題例）

- 幅が固定で十分な高さのある長方形の容器とn個の形の異なる長方形の荷物が与えられる。
- 荷物を互いに重ならないように容器内に配置する制約の下で必要な容器の高さを最小にするには？



座学編まとめ

- 量子アニーリングは、量子力学の原理を情報処理にうまく応用したコンピュータで、組合せ最適化に特化したもの。
- 既存のコンピュータでは時間がかかる特定の問題解決に大きな効果を発揮すると期待
- すべてのコンピュータが量子コンピュータに置き換わるわけではない
- 日常に組合せ最適化問題は転がっている。

実技編



量子アニーリング実践編の流れ

- ① D-Waveアカウント登録。
- ② Google Colaboratory でプログラムをアップロード
- ③ Google Colaboratory でプログラムを実行

プログラムダウンロードはこちらから

https://github.com/428lab/quantum-annealing-study-session/blob/main/001_20211213/src/quantum_annealing_study_session_001.ipynb

D-Waveマシンを実際に使ってみよう。

メールアドレスで、D-waveのアカウント登録をしよう！

<https://cloud.dwavesys.com/leap/signup/>

無料登録は・・・

- ・ 使用時間の制限が1分間（1回20 μ 秒）
- ・ 使用期間は1カ月間



FIRST NAME*

Shinya

LAST NAME*

Yamamoto

EMAIL*

q.annealing@gmail.com

LEVEL OF QUANTUM COMPUTING EXPERIENCE*

Curious person interested in learning more

JOB TITLE*

CEO

JOB DOMAIN*

Information Technology & Services

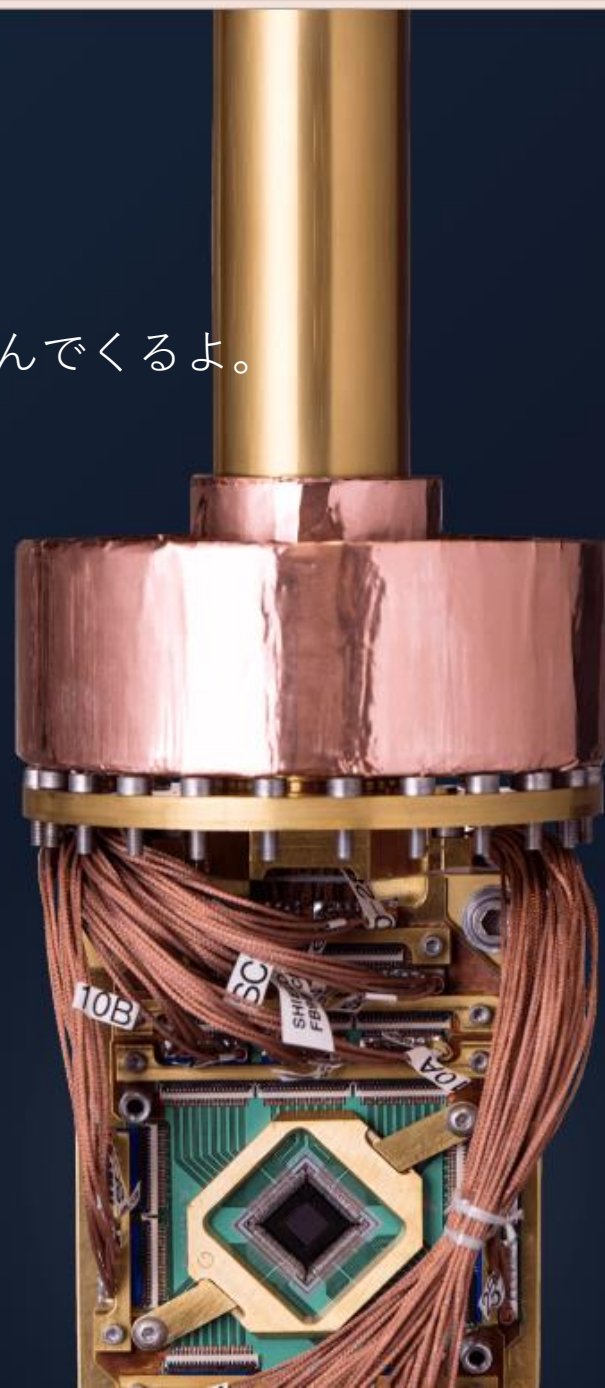
COMPANY*

Full throttle machine inc.

INDUSTRY*


Technology - Software

左記のように登録するとメールが飛んでくるよ。



アカウントを作成して、ログインするとこのようなページが見れます。
<https://cloud.dwavesys.com/leap/>

What's New



Leap Access Expanded to Singapore
February 23, 2021

S

Shinya Yamamoto

ACCOUNT TYPE

Trial Plan [GET MORE TIME](#)

TRIAL PLAN EXPIRY

October 7, 2021 (UTC)

API Token

.....

COPY

Monthly Usage Summary

Trial Plan Used
0.00%

[GET MORE TIME](#)

Usage Details

Direct QPU Total

00 h 00 m 00.000 s
TIME USED

0
PROBLEMS SUBMITTED

00 h 01 m 00.000 s
TIME AVAILABLE

Updated every minute.

Problem Status

Search by problem label

STATUS FILTER

Status of your last 1000 problems

| Problem Label | Submitted On (UTC) | Ended | Status |
|---------------|--------------------|-------|--------|
| No Results | | | |

[COPY]ボタンでTOKENを取得すれば、もう量子コンピュータを使えちゃう。プログラミングの際に使います。

Google Colaboratory

- プログラムの編集・実行をGUIで操作できるクラウド環境

<https://colab.research.google.com/notebooks/welcome.ipynb?hl=ja>

Google Colaboratory

The screenshot shows the Google Colaboratory web interface. The 'File' menu is open, and the 'Upload notebook' option is highlighted with a red box. A red text annotation is overlaid on the image, pointing to this option.

Colaboratory へようこそ

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

ノートブックを新規作成
ノートブックを開く Ctrl+O
ノートブックをアップロード
ドライブにコピー
コピーを GitHub Gist として保存
GitHub にコピーを保存
保存 Ctrl+S
変更履歴
ダウンロード
印刷 Ctrl+P

メインメニューから[ファイル]-[ノートブックをアップロード]

Colaboratory とは

Colaboratory (略称: Colab) は、ブラウザから Python を記述、実行できるサービスです。次の特長を備えています。

- 環境構築が不要
- GPU への無料アクセス
- 簡単に共有

Colab は、**学生からデータサイエンティスト、AI リサーチャー**まで、皆さんの作業を効率化します。詳しくは、[Colab の紹介動画](#)をご覧ください。下のリンクからすぐに使ってみることもできます。

はじめに

になっているこのドキュメントは静的なウェブページではなく、**Colab ノートブック**という、コードを記述して実行できるインタラクティブな環境です。

たとえば次の**コードセル**には、値を計算して変数に保存し、結果を出力する短い Python スクリプトが記述されています。

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day

86400
```

上記のセルのコードを実行するには、セルをクリックして選択し、コードの左側にある実行ボタンをクリックするか、キーボード ショートカット「command+return」または「Ctrl+Enter」を使用します。コードはセルをクリックしてそのまま編集できます。

1 つのセルで定義した変数は、後で他のセルで使用できます。

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
seconds_in_a_week
```

q_annealing.ipynb

すべて表示

Google Colaboratory



Google Colaboratory

← → ↻ colab.research.google.com/drive/18gpN94eivAGbjYFczAtKSO6_OBP8KxB?hl=ja

co q_annealing.ipynb ☆

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ 最終保存: 3:21

+ コード + テキスト

接続 編集

何はともあれ、カナダのD-Waveマシンに接続するためのライブラリをインポートします。

```
[ ] pip install dwave-ocean-sdk
```

```
Collecting dwave-ocean-sdk
  Downloading dwave_ocean_sdk-3.4.1-py3-none-any.whl (7.3 kB)
Collecting dwave-inspector==0.2.6
  Downloading dwave_inspector-0.2.6-py3-none-any.whl (25 kB)
Collecting dwave-networkx==0.8.8
  Downloading dwave_networkx-0.8.8-py2.py3-none-any.whl (81 kB)
    |#####| 81 kB 3.8 MB/s
Collecting dwave-qbsolv==0.3.2
  Downloading dwave_qbsolv-0.3.2-cp37-cp37m-manylinux1_x86_64.whl (204 kB)
    |#####| 204 kB 36.8 MB/s
Collecting penaltymodel-mip==0.2.4
  Downloading penaltymodel_mip-0.2.4-py3-none-any.whl (7.7 kB)
Collecting penaltymodel-lp==0.1.4
  Downloading penaltymodel_lp-0.1.4-py3-none-any.whl (7.3 kB)
Collecting pyqubo==1.0.12
  Downloading pyqubo-1.0.12-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.whl (887 kB)
    |#####| 887 kB 26.7 MB/s
Collecting dwave-preprocessing==0.2.0
  Downloading dwave_preprocessing-0.2.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.whl (482 kB)
    |#####| 482 kB 41.9 MB/s
Collecting penaltymodel==0.16.4
  Downloading penaltymodel-0.16.4-py3-none-any.whl (13 kB)
Collecting dwave-system==1.6.0
  Downloading dwave_system-1.6.0-py3-none-any.whl (83 kB)
    |#####| 83 kB 786 kB/s
Collecting penaltymodel-cache==0.4.3
  Downloading penaltymodel_cache-0.4.3-py3-none-any.whl (12 kB)
Collecting dwave-greedy==0.2.0
  Downloading dwave_greedy-0.2.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (655 kB)
    |#####| 655 kB 39.6 MB/s
Collecting dwave-tabu==0.4.1
  Downloading dwave_tabu-0.4.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (697 kB)
    |#####| 697 kB 39.6 MB/s
```

プログラムを正常にロードできたら、
Ctrl + F9で実行してみよう。

次回

D-Waveを使って具体的な組合せ最適化問題を解決しよう！

