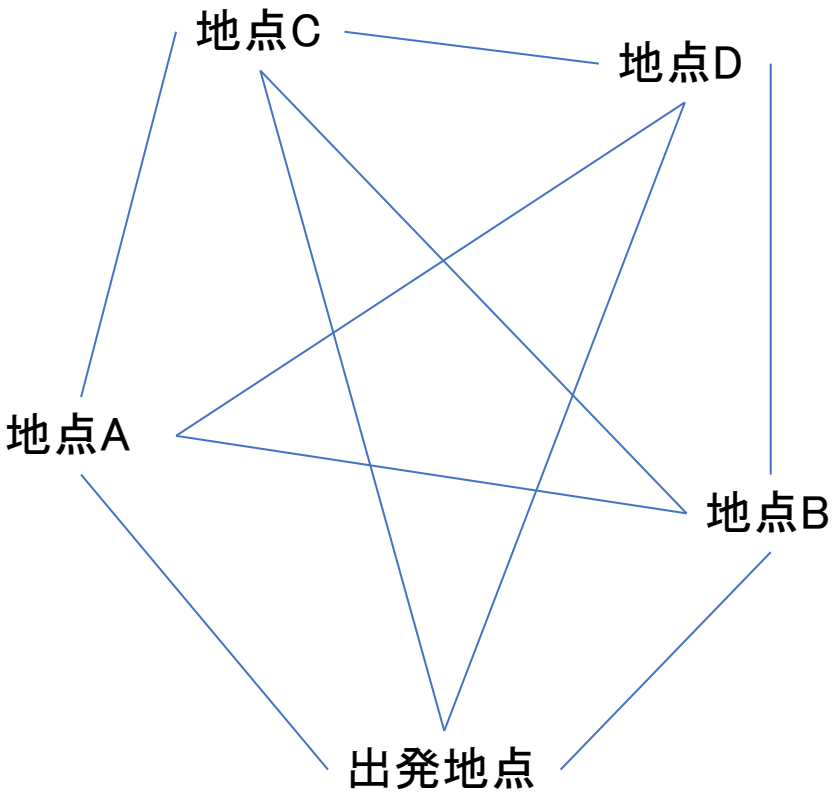
量子コンピュータ

現在解けない問題を解く

巡回セールスマン問題

出発地点からすべての地点を通り**最短**で戻ってくる

地点の数の**階乗**ものパターン数がある

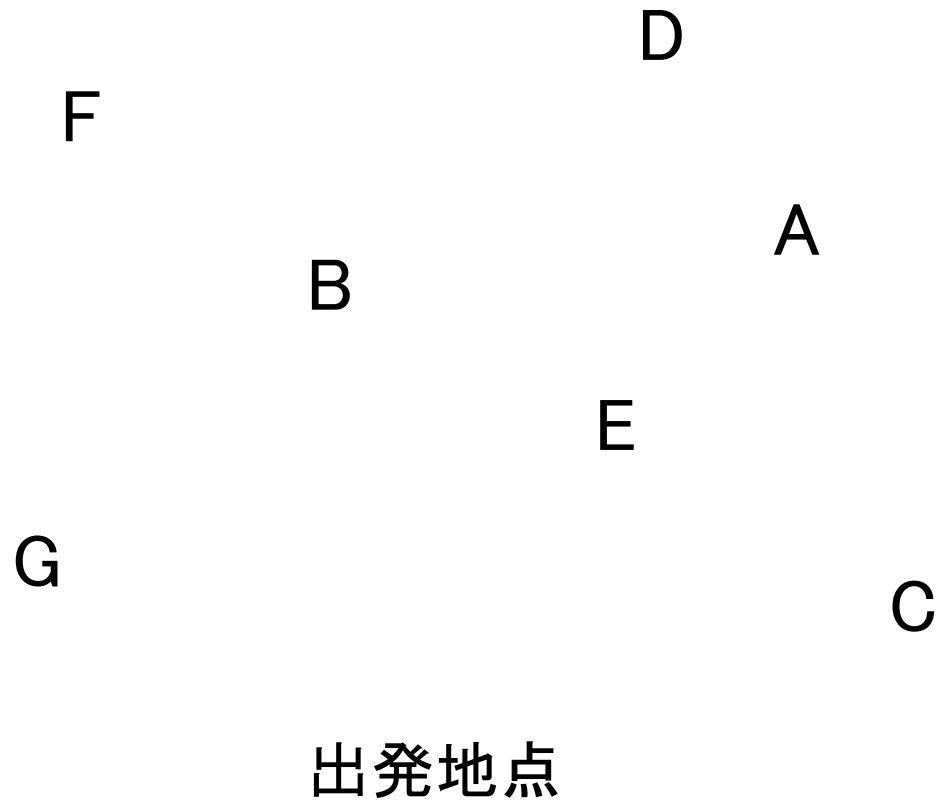


問題点

地点が増えるほど

パターン数が爆発的に増えていく

7地点だと7!=5040



実験

実際にプログラミングをして

巡回セールスマン問題を解いた

今のコンピュータでは

多くの地点を設定すると

この問題を速く解くことは難しい

結果

１０地点…約 1分

１１地点…約 10分

１２地点…約 2時間

１３地点…約 1日

１４地点…約 16日

１５地点･･･約 240日

１６地点…約 10年

量子コンピュータ

量子力学の理論を使い

1bitで0と1の両方の状態を表せるようになる

問題点

Qubitの数が少ない

最終的な答えが出せない

量子コンピュータのサイズが大きい

実験

Qubitの動きをシミュレーションできるプログラミング言語で完全乱数を作る

完全な乱数のこと

今回使用したコードの一部

```
operation Sample() : Result {
        using (q = Qubit())  {
            H(q);
            return MResetZ(q);
             }
        }
```

結果

- 完全乱数のほうは、1つの数字を出力するのに時間がかかる。
- 約100万回以上乱数を出力する場合、疑似乱数では10分もかからないが、完全乱数では約半日以上かかる。

考察

適切な量子ビットの数を設定すれば、巡回セールスマン問題を実験①より速く解けるようなプログラムになるのではないかと思う。

また、量子ビットの数によって、結果を出力する時間はかなり変わってくると思う。

展望

よりQ#への知識や理解を深め、最適化問題を解けるようなコードを書く。

量子ゲート方式、アニーリング方式のほかに使えそうなアルゴリズムを研究する。

参考文献

https://ledge.ai/quantumcomputer/

https://docs.microsoft.com/ja-jp/quantum/overview/what-is-qsharp-and-qdk