

背景

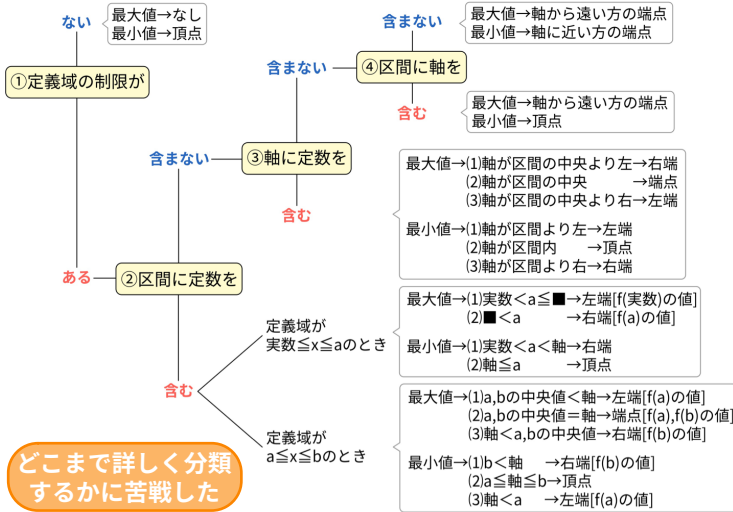
高校生に最も身近である勉強に関して、興味深い資料はないか探していると、高校生の得意・苦手教科ランキングの両方で、数学が上位に入っているのを見た。そこで、情報 I で学んで興味をもったプログラミングを利用し、数学における差がつきやすい問題の解法手順を作ってみようと思いた。

要旨

本研究では、高校数学でつまづく人が多い、2次関数の最大・最小の理解を深めるチャットボットを作成した。この分野は習い初めでは理解が困難であると感じるが、理解が深まると解法に規則性があることに気づく。この規則性をプログラミングの繰り返しを利用して組み立てられないかと考えた。チャットボットを通じて、問題の意図や目的を掴むことができ、誰でも簡単に利用できる仕組みになるよう工夫した。

作成過程

①フローチャート作成



どこまで詳しく分類するかに苦戦した

②プログラムの設計

関数の定義(def)とif文による条件分岐を主に使って作成した

高1の時にあった情報 I での学習を基に作成 1から全て作成するのは初めてで先が見えず大変だった

```
def decision_tree_dependent():
    q1 = input(条件1).lower()
    if q1 == 'yes':
        print(回答)
        q2 = input(条件2).lower()
        if q2 == 'yes':
            print(回答)
        else:
            print(回答)
    else:
        print(回答)
if __name__ == "__main__":
    decision_tree_dependent()
```

③コーディング・テスト・修正

```
print("こんにちは！2次関数の最大最小の問題を解くお手伝いをしましょう。")
print("質問に対して「はい」または「いいえ」でお答えください。")
```

```
def decision_tree_dependent():
```

```
    q1 = input(questions[0]).lower()
    if q1 == 'はい':
        q2 = input(questions[1]).lower()
        if q2 == 'いいえ':
            q3 = input(questions[3]).lower()
            if q3 == 'いいえ':
                q4 = input(questions[4]).lower()
                if q4 == 'いいえ':
                    print(advise)
                    print(hint1[4])
                    print(hint2[4])
                    print(attention)
                elif q4 == 'はい':
                    print(advise)
                    print(hint1[3])
                    print(hint2[3])
                    print(attention)
            else:
                print(response)
        elif q2 == 'はい':
            print(advise)
            print(hint1[2])
            print(hint2[2])
            print(attention)
        else:
            print(response)
    elif q1 == 'いいえ':
        q5 = input(questions[2]).lower()
        if q5 == 'はい':
            print(advise)
            print(hint1[0])
            print(hint2[0])
            print(attention)
        elif q5 == 'いいえ':
            print(advise)
            print(hint1[1])
            print(hint2[1])
            print(attention)
        else:
            print(response)
    else:
        print(response)
```

5つの質問に対して、利用者は「はい」か「いいえ」を答えるだけで解答の方針がわかる

```
#質問リスト
questions=[
    "定義域に制限がありますか?:",
    "定義域に定数を含みますか?:",
    "定義域に実数を含みますか?:",
    "軸に定数を含みますか?:",
    "定義域に軸を含みますか?:"
]
```

回答は全6通りで、教科書や問題集よりも詳細な内容となっている

50回以上のテストを行い 試行錯誤の末、自力で想像していたものを形にできた

if __name__ == "__main__": decision_tree_dependent() **完成**

実行例

問. $-1 \leq x \leq 2$ のとき、2次関数 $y = x^2 - 2x + 3$ の最大値、最小値を求めなさい。

$-1 \leq x \leq 2$ より、定義域は存在し定数は含まない。平方完成すると $y = (x-1)^2 + 2$ よって、軸は $x=1$ 、頂点は $(1, 2)$ であるから、軸は定数を含まず、定義域内に存在する。

こんにちは！2次関数の最大最小の問題を解くお手伝いをしましょう。質問に対して「はい」または「いいえ」でお答えください。

```
定義域に制限がありますか?:はい
定義域に定数を含みますか?:いいえ
軸に定数を含みますか?:いいえ
定義域に軸を含みますか?:はい
~その場合はこのようにお考え下さい!~
最大値 - 軸から遠い方の端点
最小値 - 頂点
これは下に凸の場合です。問題が上に凸の場合は「最大値」と「最小値」を入れ替えて考えてくださいね!
```

オレンジ色で囲われているところが利用者に対する質問で、これらは実際の問題から簡単に読み取ることができる

結果

2次関数の最大・最小を求める問題を解く方法を示すプログラムを作成できた。しかし、今回プログラムを作成するにあたって、言葉の定義の危うさや曖昧さ、他者との認識の違いを知った。また、いかに自分が理解していないかを理解し、これからは物事の本質を捉えられるように心掛けたいと思った。

今後の展望

- ツールやアプリとして開発していく
- 実際に活用してもらいアンケートを取って改良する
- 文字ではなく視覚的に分かりやすい図やグラフで回答を表示する
- 基本語句の意味を確認できる一覧表を作成する
- 2次関数以外の単元にも応用する