

コーディング課題

問1

実行方法(srcディレクトリ下にq1_in.txtを置いてください)

```
$ python src/q1.py
```

解法

xとyが2通りでいずれかが重複を許して選ばれるから、全部で512(2の9乗)組ある。これらの行列式を求めてdと一致するような値を数えればいい。dと一致すればTrue、不一致ならFalseなのでこれらの総和を取れば数えた場合と同じ結果になる。

検証方法

まず3×3の重複順列になっているかをネストしたfor文と比較する。次に同じ値の行列式の数分かっている行列を入力として、dと一致する総和が等しいことを確認する。最後にテストした2つの関数を結合した関数が、ネストしたfor文の行列式のdの値の数が一致することを確認する。

テスト実行方法

```
$ python src/test_q1.py
```

問2

実行方法(srcディレクトリ下にq2_in.txtを置いてください)

```
$ python src/q2.py
```

解法

S_k は次のような規則性がある。 $S_k = S_{k-3} + S_{k-2} + S_{k-1}$

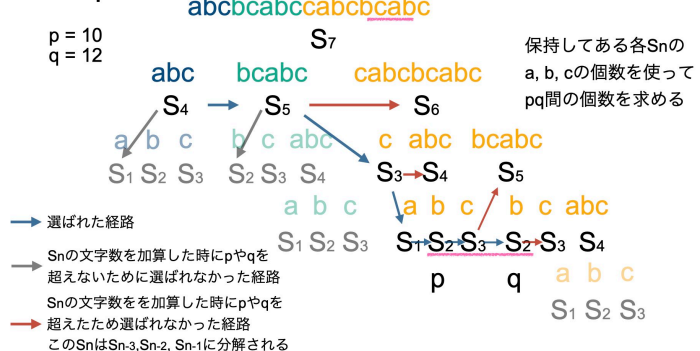
分解した文字列 $S_{k-i}(i \geq 0)$ は $k-i \geq 4$ ならばさらに分解し続けることができる。加えて、最小まで分解すれば S_1 と S_2 と S_3 のいずれかから構成されることがわかる。

また、 $k > 5$ のとき S_{k-3} は S_{k-2} や S_{k-1} の構成要素でもある。そのため S_{k-3} の文字数がわかれば、 S_{k-2} の文字数も S_{k-3} の計算結果をそのまま利用できる。したがって、ある $S_n(1 \leq n \leq k)$ でのabcの個数を保持すれば、計算効率いいことがわかる。

pq間のabcの個数を数えるときは保持している各 S_k でのabcの個数が $S_n(1 \leq n \leq k)$ においてpq間を超えるかを確認しながら探索する。

pq間を超えるときは、超えなくなるまで S_n を分解。pq間を超えないときは、 S_n を分解せずにpやqに達するまで保持している総数(S_n の文字列)を足し続ける。pq間に達したら、a,b,cを数え始める。qと一致したら数えるのを終了する。

Example



検証方法

問題文の通りに文字列Sk-3とSk-2とSk-1を足したもののpq間をスライスで取り出してabcの個数を数え、作成した関数の結果と一致するかを確認する。kが大きいときはテスト時間が遅いことと、漸化式なのでその後も同様の処理が行われると考え、k=25までの個数の一致をもって正しいものであると判断した。

テスト実行方法

```
$ python src/test_q2.py
```

問3

実行方法(srcディレクトリ下にq3_in.txtを置いてください)

```
$ python src/q3.py
```

解法

既に着席済みの席と席の間(以後区間とする)に座る時を考える。この区間が奇数なら、中心は両側の座っている席から同じくらい離れているので中心に座る。偶数なら中心より左と右はどちらも座っている席から同じくらい離れているが、条件より中心より左に座る。区間が等しい箇所が複数存在する時、離れている距離は変わらないため、条件より一番左の区間から座る席を決める。区間内の着席済みの席との離れている長さは区間の長さに依存する。そのため、区間が一番大きい時はその区間の中心の席が選ばれる。

次に端の席が空いている時を考える。この時、aから端までの長ささと区間の中心の長さを比べる。左端が空いている場合は中心までの長さ以上であれば左端が選ばれる。右端は中心までの長さより大きい時に選ばれる。

以上より、席と席との区間の長さが必要である。加えて、長さが等しいときは左側を選択するため、長さの優先順位が必要になる。長さの並び替えは標準ライブラリの最小ヒープを使う。最小ヒープなので、長さを降順にするために負の値変換する。長さが等しいときはindexを使って低いindexからなる長さを取り

出せるようにする。これを行うにはタプル(-length, index)で格納すればいい。また、両端をヒープから取り出せるように、初めに長さを二倍して1引いて格納する。そのためヒープから取り出せた後は元の長さに戻す必要がある。これらを判別するため取り出したindexが端になるかをif文で確認する。後は順にヒープで取り出してaを求めていけばいい。

検証方法

長さが奇数のときと偶数のときの選ばれる位置は正しいか。長さが複数ある時に一番長いものを選ぶか。長さが等しいときは一番左から選ぶか。実際にaが与えられた時に自らが考える結果になっているか。これらを確認すれば良い。

テスト実行方法

```
$ python src/test_q3.py
```

