

5次魔方陣の取り組みについて

— 広島大学理学部中・高校生科学シンポジウムにむけて —

甲斐 章義

2014年2月に茨城県の高校生が筑波大学のスーパーコンピュータ「T2K-Tsukuba」を用いて5次魔方陣2億7530万5224通りの全解を2時間36分かかって求めたことがインターネット上でニュースになっていた。プログラムは「枝刈り法」と呼ばれる手法であったことも一緒に紹介されていた。以前、この「枝刈り法」とほとんど同じ発想でプログラムを作成し通常のパソコンで実行した所、莫大な時間を要することがわかって途中で断念した経験と、まったく異なるアルゴリズムの着想が頭にあったことから、そのアルゴリズムをプログラムに書き起こし実行してみることにした。そこで生徒とともに協力して情報処理演習室のパソコン44台の並列計算を行うことで、より短い時間で5次魔方陣の全解を求めるプロジェクトを立ち上げた。その活動の様子は11月始めに広島大学理学部で開かれる中学生・高校生科学シンポジウムで発表したの、その内容をここに報告したい。

1. はじめに

プログラムの試作は筑波大学のニュースがインターネット上で流れた直後から行った。プログラム言語にはpythonを使用した。pythonはLL言語であるため実行速度は遅いが、扱いやすい言語であることと、私自身がpythonを学習中であったことからpythonを採用することにした。試作版を含めプログラムがまともに動くまで2ヶ月を要した。

一方でこの活動をともに行う生徒は基本的にコンピュータ同好会の生徒に協力してもらった。コンピュータ同好会にはコンピュータ言語としてC言語、C++、pythonなど生徒によって様々な言語に取り組んでいる。今回はその中からpythonに取り組んでいる生徒に声をかけた。理由はもちろん私が作成した試作版プログラムがpythonで書かれているからである。生徒の取り組みとしては、まず5次魔方陣の基本的な性質の学習から始めてもらった。5次魔方陣は回転・対称移動などで8個の5次魔方陣が同一視できる。5次魔方陣2億7530万5224通りはこれらを同一視してカウントした数である。さらに5次魔方陣には同型写像が存在し、4個の魔方陣が同一視できる。したがって、これらの同型写像を意識すると数えないといけない5次魔方陣は6882万6306通りであることがわかる。次に試作したプログラムのアルゴリズムを理解することに取り組んでもらった。プログラムを改良し並列計算をすることができるように変更するところについては生徒に取り組んでもらうつもりであったからである。

2. アルゴリズムについて

プログラムのアルゴリズムについて簡単に説明する。5

次魔方陣は5×5の25個のマスに1から25までの数を入れて縦・横・斜めの和がすべて同じになるようにする物である。プログラムの都合上、マスに入れる数を1から25ではなく0から24までとする。0から24までの総和は300なので、縦・横・斜めの和は60になればよい。

プログラム内ではこれら25個の数を1列目から順に並べて25桁の25進数と見なす。この25進数を小さい順にチェックし5次魔方陣の総数をカウントしていくのが基本的な考え方である。ただ、まともに全ての順列をチェックしてはとても時間が足りないので次のような工夫をする。

①横の和を60にする。

まず1行目の5個を昇順でその和が60になるように並べる。2行目は残りの数のうちの5個を昇順でその和が60になるように並べる。以下、3行目、4行目と繰り返すと5行目は昇順にすると自動的にその和は60となる。こうして5個の横の和が60である25桁の25進数ができあがる。このとき各行を5桁の25進数と見なすとこれら5個の25進数もまた昇順に並ぶ。これを25桁の25進数として昇順に順次作成していく。例えばこの段階で最初に出てくるのは

0	1	12	23	24
2	3	13	20	22
4	5	11	19	21
6	7	14	15	18
8	9	10	16	17

である。

②縦の和を60にする。

①で作成したものの1行目は固定したままにする。2行目以降はその行の中で並び替えを行い、1列目の縦の

和が 60 になるように並び替える。その行の中での並び替えなので各行の和は 60 のまま変わらない。この操作を 2 列目, 3 列目, 4 列目と繰り返していくと自動的に 5 列目の和も 60 となる。

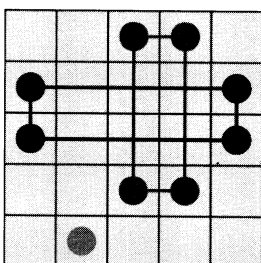
例えば①の例に対してこの操作を行った結果が

0	1	12	23	24
13	22	20	2	3
19	21	5	4	11
18	7	6	15	14
10	9	17	16	8

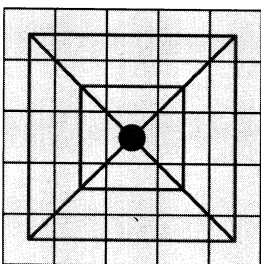
となる。

③斜めの和を 60 にする。

横と縦の和がともに 60 となっているので、行どうしまたは列どうしを入れ替えてもやはり横と縦の和はともに 60 のままである。したがって行どうし列どうしの入れ替えをすることで斜めの和を 60 にする。まず各行から 1 個ずつ数を選びその和が 60 となる組み合わせのリストを作る。そのリストから共通項が 1 個のみで、残りの 4 個ずつで長方形が 2 個できるものを選ぶ。(例えば下図)



これを行どうし列どうしの入れ替えで変形すると



となり 5 次魔方陣が完成する。5 次魔方陣の個数をカウントするだけであれば最後の変形は必要がないので、その分だけ実行速度は速くなる。

以上①, ②, ③の操作を再帰的に昇順で繰り返して魔方陣を作成していく。

実際のプログラム

```
#!/usr/bin/python
#-*- coding: utf-8 -*-

import datetime

def sort(c=[i for i in range(25)],k=1):
    for i in range(k,25):
        for j in range(i+1,25):
            if c[i]>c[j]:
                temp=c[i]
                c[i]=c[j]
                c[j]=temp
    return c

def sort5(x=[i for i in range(5)],k=1):
    for i in range(k,5):
        for j in range(i+1,5):
            if x[i]>x[j]:
                temp=x[i]
                x[i]=x[j]
                x[j]=temp
    return x

def inc(c=[i for i in range(25)],k=23,l=1):
    if c[k]+l>=25:
        c = inc(c=c,k=k-1,l=1)
        c=sort(c=c,k=k)
        return c
    else:
        re=0
        for i in range(k+1,25):
            if c[k]+l == c[i]:
                d = c[k]
                c[k] = c[i]
                c[i] = d
                if k<=22:
                    c=sort(c=c,k=k+1)
                re=1
                break
        if re==0:
            return inc(c=c,k=k,l=l+1)
        else:
            return c

def inc5(x=[i for i in range(5)],k=3,l=1):
    if k<0:
        return [0 for i in range(5)]
    elif x[k]+l>=5:
        x=inc5(x=x,k=k-1,l=1)
        x=sort5(x=x,k=k)
        return x
    else:
        re=0
        for i in range(k+1,5):
            if x[k]+l == x[i]:
                d = x[k]
                x[k] = x[i]
                x[i] = d
                if k<=2:
                    x=sort5(x=x,k=k+1)
                re=1
                break
        if re==0:
            return inc5(x=x,k=k,l=l+1)
        else:
            return x
```

```

def replace(c=[i for i in range(25)]):
    i=0
    while i<5:
        re=-1
        if i==0:
            while c[0]+c[1]+c[2]+c[3]+c[4]<>60 or c[0]>c[1] or
c[1]>c[2] or c[2]>c[3] or c[3]>c[4]:
                if c[0]>c[1] or c[0]+c[1]+c[2]*3>57:
                    c=inc(c=c,k=1)
                elif c[1]>c[2] or c[0]+c[1]+c[2]+c[3]*2>59:
                    c=inc(c=c,k=2)
                elif c[2]>c[3]:
                    c=inc(c=c,k=3)
                else:
                    c=inc(c=c,k=4)
            elif i==1:
                while c[5]+c[6]+c[7]+c[8]+c[9]<>60 or c[5]>c[6] or
c[6]>c[7] or c[7]>c[8] or c[8]>c[9] or c[0]>c[5]:
                    if c[0]>c[5] or c[5]+c[6]*4>54:
                        c=inc(c=c,k=4)
                    elif c[5]>c[6] or c[5]+c[6]+c[7]*3>57:
                        c=inc(c=c,k=6)
                    elif c[6]>c[7] or c[5]+c[6]+c[7]+c[8]*2>59:
                        c=inc(c=c,k=7)
                    elif c[7]>c[8]:
                        c=inc(c=c,k=8)
                    else:
                        c=inc(c=c,k=9)
                while
c[5*i]+c[5*i+1]+c[5*i+2]+c[5*i+3]+c[5*i+4]<>60 or
c[5*i]>c[5*i+1] or c[5*i+1]>c[5*i+2] or c[5*i+2]>c[5*i+3] or
c[5*i+3]>c[5*i+4] or c[5*(i-1)]>c[5*i]:
                    if c[5*(i-1)]>c[5*i] or c[5*i]+c[5*i+1]*4>54:
                        c=inc(c=c,k=5*i-1)
                    elif c[5*i]>c[5*i+1] or
c[5*i]+c[5*i+1]+c[5*i+2]*3>57:
                        c=inc(c=c,k=5*i+1)
                    elif c[5*i+1]>c[5*i+2] or
c[5*i]+c[5*i+1]+c[5*i+2]+c[5*i+3]*2>59:
                        c=inc(c=c,k=5*i+2)
                    elif c[5*i+2]>c[5*i+3]:
                        c=inc(c=c,k=5*i+3)
                    else:
                        c=inc(c=c,k=5*i+4)
            if i>=1:
                if c[0]>c[5] or c[0]+c[1]+c[2]+c[3]+c[4]<>60:
                    re=0
                else:
                    for j in range(i):
                        if j==0:
                            if c[0]>c[1] or c[1]>c[2] or c[2]>c[3] or
c[3]>c[4] or c[0]+c[1]+c[2]+c[3]+c[4]<>60:
                                re=0
                                break
                            else:
                                if c[5*j-5]>c[5*j] or c[5*j]>c[5*j+1] or
c[5*j+1]>c[5*j+2] or c[5*j+2]>c[5*j+3] or c[5*j+3]>c[5*j+4] or
c[5*j]+c[5*j+1]+c[5*j+2]+c[5*j+3]+c[5*j+4]<>60:
                                    re=j
                                    break
                        if re==-1:
                            i+=1
                        else:
                            i=re
    return c

```

```

def replace2(c=[i for i in range(25)],k=0):

```

```

    a=[[0 for j in range(5)]for i in range(5)]
    for i in range(5):
        for j in range(5):
            a[i][j]=c[5*i+j]
    k=colreplace(a=a,k=k)
    return k

```

```

def colreplace(a=[[0 for j in range(5)]for i in range(5)],s=0,k=0):
    r=5*s
    i=1
    t=[0 for j in range(5)]
    while i<r**4:
        b=i
        for j in range(4):
            t[4-j]=b%r
            b=int(b/r)
        if
a[0][s+a[1][s+t[1]]+a[2][s+t[2]]+a[3][s+t[3]]+a[4][s+t[4]]==60:
            for j in range(1,5):
                temp=a[j][s+t[j]]
                a[j][s+t[j]]=a[j][s]
                a[j][s]=temp
                a[j]=sort5(x=a[j],k=s+1)
            if s>=3:
                if a[0][1]<a[1][0] and a[0][1]<a[2][0] and
a[0][1]<a[3][0] and a[0][1]<a[4][0]:
                    k=crossreplace(z=a,k=k)
                for j in range(1,5):
                    a[j]=sort5(x=a[j],k=s)
            else:
                k=colreplace(a=a,s=s+1,k=k)
                for j in range(1,5):
                    a[j]=sort5(x=a[j],k=s)
        i+=1
    return k

```

```

def transfer(c=[[0 for j in range(5)] for i in range(5)]):
    txt='0123456789abcdefghijklmno'
    t=""
    for i in range(5):
        for j in range(5):
            t+=txt[c[i][j]]
    return t

```

```

def crossreplace(z=[[0 for j in range(5)]for i in range(5)],k=0):
    lst=crosslist(z=z)
    t=""
    length=len(lst)
    d=[[0 for j in range(5)]for i in range(5)]
    x=[0 for i in range(5)]
    y=[0 for i in range(5)]
    if length>=2:
        for i in range(length-1):
            for j in range(i+1,length):
                for m in range(5):
                    for n in range(5):
                        d[m][n]=z[m][n]
                for m in range(5):
                    x[m]=lst[i][m]
                    y[m]=lst[j][m]
                count=0
                num=-1
                for m in range(5):
                    if x[m]==y[m]:
                        count+=1
                        num=m
                if count==1:
                    v=x[num]

```

```

if v!=2:
    ik=0
    while ik<5 or x[num]!=2:
        if x[ik]==2:
            x[ik]=v
            x[num]=2
            ik+=1
        ik=0
        while ik<5 or y[num]!=2:
            if y[ik]==2:
                y[ik]=v
                y[num]=2
                ik+=1
        d=colswich(d=d,m=2,n=v)
if num !=2:
    x[num]=x[2]
    x[2]=2
    y[num]=y[2]
    y[2]=2
    d=rowswich(d=d,i=2,j=num)
if x[0]+y[0]==1 or x[0]+y[0]==7:
    n1=0
    n2=0
    for ik in range(5):
        if x[ik]==0:
            n1=ik
        if x[ik]==3:
            n2=ik
    temp=x[n1]
    x[n1]=x[n2]
    x[n2]=temp
    n1=0
    n2=0
    for ik in range(5):
        if y[ik]==0:
            n1=ik
        if y[ik]==3:
            n2=ik
    temp=y[n1]
    y[n1]=y[n2]
    y[n2]=temp
    d=colswich(d=d,m=0,n=3)
elif x[0]+y[0]==3 or x[0]+y[0]==5:
    n1=0
    n2=0
    for ik in range(5):
        if x[ik]==0:
            n1=ik
        if x[ik]==1:
            n2=ik
    temp=x[n1]
    x[n1]=x[n2]
    x[n2]=temp
    n1=0
    n2=0
    for ik in range(5):
        if y[ik]==0:
            n1=ik
        if y[ik]==1:
            n2=ik
    temp=y[n1]
    y[n1]=y[n2]
    y[n2]=temp
    d=colswich(d=d,m=0,n=1)
if x[0]+y[0]==4 and x[1]+y[1]==4 and
x[3]+y[3]==4 and x[4]+y[4]==4:
    for ik in range(5):
        if x[ik] !=ik:
            jk=idx(x=x,r=ik)
            x[jk]=x[ik]
            x[ik]=jk
            d=rowswich(d=d,i=ik,j=jk)
            k+=1
            t+=transfer(c=d)
    fl.write(t)
    return k
def idx(x=[i for i in range(5)],r=0):
    v=-1
    for i in range(5):
        if x[i]==r:
            v=i
            break
    return v
def rowswich(d=[[0 for j in range(5)]for i in range(5)],i=0,j=1):
    temp=d[i]
    d[i]=d[j]
    d[j]=temp
    return d
def colswich(d=[[0 for j in range(5)]for i in range(5)],m=0,n=1):
    temp=[0 for i in range(5)]
    for i in range(5):
        temp[i]=d[i][m]
    for i in range(5):
        d[i][m]=d[i][n]
    for i in range(5):
        d[i][n]=temp[i]
    return d
def crosslist(z=[[0 for j in range(5)]for i in range(5)]):
    lst=[]
    x=[i for i in range(5)]
    while x[0]+x[1]>0:
        if
z[0][x[0]]+z[1][x[1]]+z[2][x[2]]+z[3][x[3]]+z[4][x[4]]==60:
            y=[x[0],x[1],x[2],x[3],x[4]]
            lst.append(y)
            x=inc5(x=x,k=3,l=1)
    return lst
if __name__ == "__main__":
    now=datetime.datetime.now()
    print "%d/%d/%d" %d:%d:%d%(now.year,now.month,now.day,now.hour,now.minut
e,now.second)
    f=open("record.txt","w")
    f.write("start %d/%d/%d %d:%d:%d"
"%(now.year,now.month,now.day,now.hour,now.minute,now.seco
nd))
    f.close()
    fl=open('mdata.dat','w')
    counter=0
    c = [i for i in range(25)]
    d = [i for i in range(25)]
    ここに初期条件を記入
    c=replace(c=c)
    while c[1]<12:←終了条件
        for i in range(25):
            d[i]=c[i]
            counter=replace2(c=c,k=counter)
            if c[5]!=d[5] or c[6]!=d[6] or c[7]!=d[7] or c[8]!=d[8] or
c[9]!=d[9]:
                print c[0],c[1],c[2],c[3],c[4]
                print c[5],c[6],c[7],c[8],c[9]," counter=",counter
                c=inc(c=c,k=19)
                c=replace(c=c)

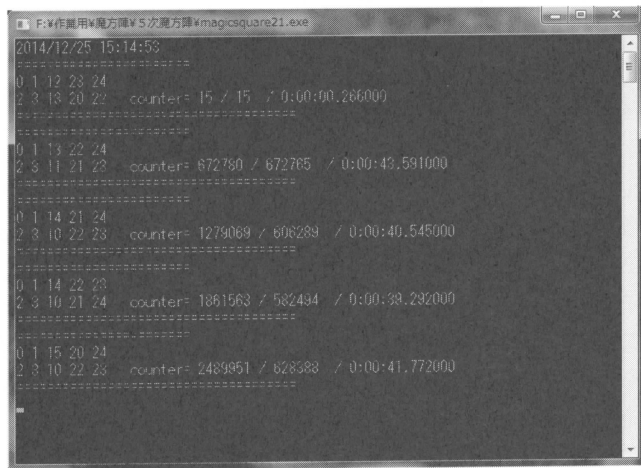
```

```

now=datetime.datetime.now()
print "%d/%d/%d" % (now.year,now.month,now.day,now.hour,now.minut
e,now.second)
f.close()
f=open("record.txt","a")
f.write("counter=%d "%counter)
f.write("finish %d/%d/%d"
%d:%d:%d"%(now.year,now.month,now.day,now.hour,now.minut
e,now.second))
f.close()

```

プログラムの実行画面



3. プログラムの実行方法

プログラム自体は python で書かれている。25 桁の 25 進数を昇順に検索しているので、検索の始まり（プログラム中の初期条件の部分指定する）と終わり（プログラム中の終了条件を指定する）を指定してやれば複数のパソコンで並列計算をさせることもたやすい。パソコンどうして同期をとるようなことはせずにプログラムをコンピュータの数だけ作り、同時に実行することで時間の短縮を図ることにした。

もう一つの問題は python の実行速度の遅さである。pyhton のままで上記プログラムを実行すると約 34 時間 30 分かかった。スーパーコンピュータ「TK2-Tsukuba」の 512CPU (4.7TFLOPS 相当) を用いて 2 時間 36 分かかったことを考慮すれば一般のパソコンで 34 時間 30 分は速いかもしれないが、インターネット上では同じ計算を一般のパソコンで 10 分で行ったという話題もあり、少なくともこの 10 分は切りたい。また机上の計算ではあるが、プログラムの高速化を図り、校内の情報処理演習室にあるコンピュータ 44 台すべてを使用した並列計算を行うと 45 秒前後、5 次魔方陣の個数だけカウントするのであれば 30 秒強で計算が可能であるとの試算もあった。したがってこのレベルを目指すことにした。

プログラムの高速化については Cython を利用して C 言語でコンパイラする方法と shedskin というソフトで C++に変換後ビルドして exe ファイルを作る方法の 2 通りを検討した。結果として扱いやすく計算速度も速かった後者を選択した。

プログラムの並列化についてはこれは全面的に生徒に任せた。最初の計算でどの領域でどの程度計算時間がかかるのかが実測できていたので、それを参考に情報処理演習室のコンピュータ 44 台、4 core の CPU なので、全領域を 176 個に分割して計算するために領域ごとにプログラム 176 個を作成した。

4. プログラムの実行

実際にプログラムを実行する際には領域の設定ミスでプログラムの個数が 178 個となったが、そのまま 178 個で実行することにした。最初は 178 個のプログラムをサーバー内に置き実行したが、算出した 5 次魔方陣を保存する際にサーバー内のファイルに保存するため、サーバーへのアクセスが多くなりすぎ、それがボトルネックとなって思うほど計算速度が上がらなかった。そこで 178 個を 4 個ずつ 43 台に配置し、残り 6 個は教室前にあるコントロール機に配置して、手作業で exe を実行した。実行結果は最初の目論見通りで、5 次魔方陣の個数のみをカウントする方法では 35 秒、5 次魔方陣を算出し保存する方でも 44 秒で計算することができた。

5. シンポジウムでの発表

広島大学では毎年 11 月初旬に広島大学理学部・大学院理学研究科公開中学生・高校生科学シンポジウムを開催しており、今年で 17 回目となる。我が校からも毎年のように研究発表を行っており、今年も 5 次魔方陣の研究発表とエコクラブによる植物の成長と鉄との関係の発表が行われた。口頭発表は各校から 1 個のみであとはポスター発表となっており、5 次魔方陣については口頭で、エコクラブの方はポスターで発表が行われた。

5 次魔方陣の取り組みは口頭発表なので、そのためのプレゼンテーションの作成と発表原稿の作成、および発表資料の作成の 3 点に取り組んだ。プレゼンテーションだけではプログラムやアルゴリズムの詳細がわからないので発表の際の補助資料として発表資料を作成した。

まずはパワーポイントでプレゼンテーションを作成し、発表の際に使用する発表原稿の作成を担当する係と発表の補助資料となる発表資料の作成する係とに生徒を分けて、期日を設けて作成させた。プレゼンテーションと発表原稿、および発表資料ができあがったところで、発表の予行演習を行い、プレゼンテーションと発表原稿、

発表資料との相違点や順序の違いなどを洗い出し、よりわかりやすい発表となるように生徒どうしで話し合わせ、プレゼンテーションや発表原稿、発表原稿の修正を行った。特に発表原稿の修正は発表当日の発表ぎりぎりまで行われた。その結果、プレゼンテーションはスライド15枚、発表原稿はA4で3ページ、発表資料はアルゴリズムの解説やプログラム本体、計算領域の分割の様子、プログラムの実行結果の一覧など19ページにおよぶものになった。

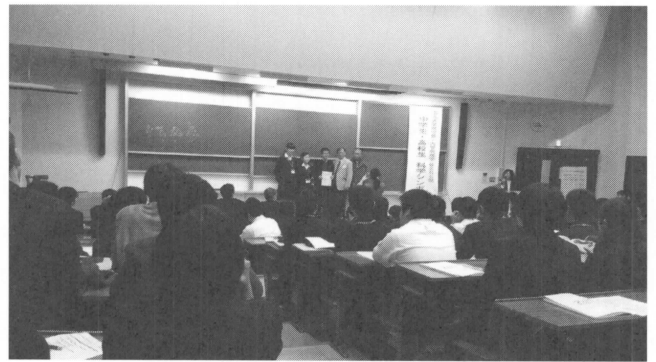
シンポジウムでの口頭発表は当日の午後に行われ、我が校の発表はその最後であった。準備の甲斐があっただけに口頭発表の制限時間8分以内で無事発表を終えることができた。筑波のスーパーコンピュータで2時間36分かかった計算を通常のパソコン44台を使用して44秒で計算したということで、発表を聞いていた他の学校の生徒やコメントのために臨席していただいていた大学の先生からも多大な賞賛の声をいただいた。



シンポジウムで発表している様子



科学奨励賞授賞式での賞状授与の様子



授与式後の記念写真撮影の様子

6. 生徒の感想

シンポジウムの取り組みを終えて、生徒はとても感激した様子であった。特にこの取り組みを大学の先生に絶賛されたことについてはいたく感激していた。また、校外に出て他校の生徒の取り組みの発表を聞いたり、また自らの取り組みを発表したりする機会はそうそうないのでとても勉強になったし、他の学校の生徒がどんな取り組みをしているのかがよくわかり勉強になったという感想を述べていた。また、大学の先生の計らいでシンポジウム後に、大学の研究室を見学させていただくことができたこともとても良い経験であったようである。