

知乎

首发于
次元壁

分享: 一段老代码的改进



酱紫君

数学 话题的优秀回答者

已关注

48 人赞同了该文章

最近在写一个项目整理一些代码

这是一段算24点的代码 douban.com/group/topic/...

```
filter[{op1_, op2_, op3_}, {a_, b_, c_, d_}, t_:24]:=Module[
  {ans={}},
  If[Quiet@op1[op2[op3[a,b],c],d]==24,ans=Union[ans,{HoldForm@op1[op2[op3[a,b],c],d]}],
  If[Quiet@op1[op2[a,b],op3[c,d]]==24,ans=Union[ans,{HoldForm@op1[op2[a,b],op3[c,d]}],
  If[Quiet@op1[op2[a,op3[b,c]],d]==24,ans=Union[ans,{HoldForm@op1[op2[a,op3[b,c]],d]}],
  If[Quiet@op1[a,op2[op3[b,c],d]]==24,ans=Union[ans,{HoldForm@op1[a,op2[op3[b,c],d]}],
  If[Quiet@op1[a,op2[b,op3[c,d]]]==24,ans=Union[ans,{HoldForm@op1[a,op2[b,op3[c,d]]}],
  ans];

cal24[nums_List]:=Module[
  {ops,num=nums,rule,same},
  ops={Plus,Subtract,Times,Divide};
  rule=Thread[num->{n1,n2,n3,n4}];
  same[a_,b_]:=Module[{p,q},p=a/.rule;
    q=b/.rule;ReleaseHold[p]==ReleaseHold[q];
    DeleteDuplicates[Union@@filter[ops,nums,rule,same,t_]]];
```

▲ 赞同 48



● 10 条评论

➦ 分享

★ 收藏

知乎

首发于
次元壁

这段代码比其他我找得到的最快的代码还要快5倍(1秒20个模式,一分多钟检验完24点所有的可能性), 且不重不漏.

(不得不计算24点的代码很多都不做去重...最难的部分就是去重)

简单点评下这段代码, 主要就是最后一行

```
DeleteDuplicates[Union@@filter@@@Tuples[{Tuples[ops,3],Permutations[num]}]]
```

直接生成所有模式, 用filter判定, 因为算式都是HoldForm, 于是直接取并集(Union)去重即可.

那么问题来了...这个filter怎么写出来的...五个If 哪来的?

我问了下吧主...吧主表示...

7年前的代码谁还记得...

大概是手写归纳的,因为原文中还漏掉了一个模式...

然后大佬 @曹洪洋 出手大大化简了这段代码.

```
calc24[nList_]:=DeleteDuplicatesBy[Cases[  
  Outer[filter,Tuples[{Plus,Subtract,Times,Divide},3],Permutations[nList],1],e_  
  ReleaseHold[#/.Thread[nList->{x,y,z,w}]]&];
```

把 Quiet@ReleaseHold@e==24 给移出来了.

这下难度就不大了...

但是问题还在于filter到底是什么

我们画出树状图

▲ 赞同 48

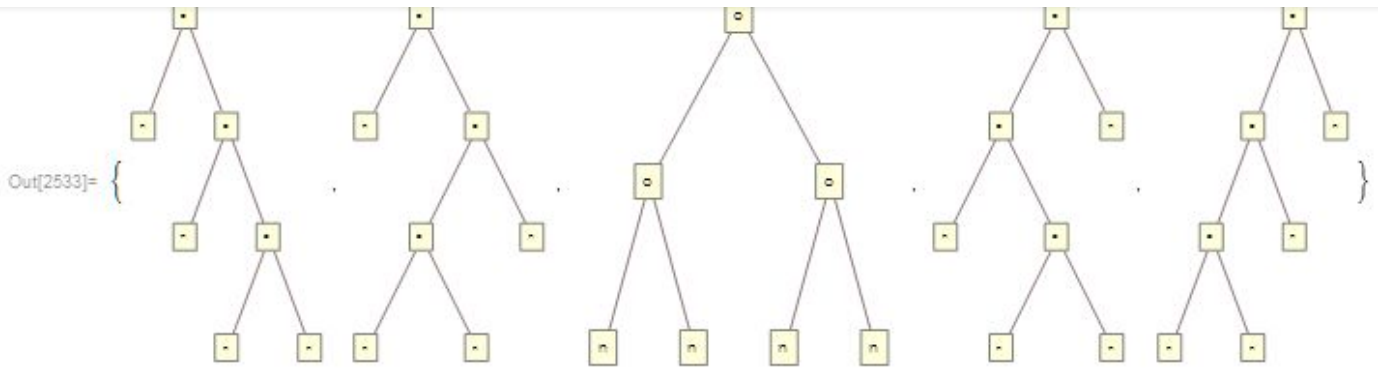


● 10 条评论

➤ 分享

★ 收藏

知乎

首发于
次元壁

哇,n个节点的有序有根二叉树...或者叫卡特兰树...

因为树的数目是卡特兰数.

于是我们就能用元编程解决这个问题了

```
treeR[1]=n;
treeR[n_]:=treeR[n]=Table[o[treeR[a],treeR[n-a]],{a,1,n-1}]
treeC[n_]:=Flatten[treeR[n]//.{o[a_List,b_]:>(o[#,b]&/@a),o[a_,b_List]:>(o[a,#{b_}])]&
treeC[4]//TableForm

nn=Array[ToExpression["n"<>ToString@#]&,4]
oo=Array[ToExpression["o"<>ToString@#]&,4-1]
ff=ReplacePart[#,Thread[Position[#,n]->nn]~Join~Thread[Position[#,o]->oo]]&
filter=Function[Evaluate@Join[oo,nn],Evaluate[HoldForm/@Evaluate[ff/@treeC[4]]]]

calc24[nList_]:=DeleteDuplicatesBy[Cases[Outer[filter@@Join[#1,#2]&,Tuples[{Plus,Minus,Multiply,Divide}],{nList}]]
calc24[{2,3,5,7}]
```

算法规模是 $f_o(n) = n!C_{n-1}o^{n-1}$

C是卡特兰数,o是操作符数,这里是加减乘除4个

$$f_4(n) = O\left(\left(\frac{16}{e}\right)^n \left(n - \frac{3}{2}\right)^{n-1}\right)$$

发布于 2017-12-21

Wolfram Mathematica

24 点

数学

▲ 赞同 48



● 10 条评论

➤ 分享

★ 收藏



文章被以下专栏收录



次元壁

介绍一些有关数学或者游戏的有趣内容,转载不需要授权,表明作者和出处就行....

关注专栏

推荐阅读

数学新进展：数学家证明了一个重要的不等式

本文转载自百家号：量子认知链接：数学新进展：数学家证明了一个重要的不等式俄罗斯数学家最近证明了一个重要的数学不等式。这个成果具有重要的数学和物理学意义。所提供的新的数学工具可以...

数学经纬网

用HoTT/CuTT玩转代数拓扑：证明 $\text{torus} = S^1 \times S^1$

现在主流HoTT的可计算模型都基于cubical type,而ct又有2种：cartesian 与 De Morgan：而今天介绍的是一个基于cartesian cubical set 的theorem prover : redtt。这里是一个HIT的例子：在代数...

Marti...

发表于algeb...

10 条评论

⇌ 切换为时间排序

写下你的评论...



芷优生活

2 年前

哇塞，顶一个

👍 赞



wxway

2 年前

哇，好厉害!!!

👍 赞



wxway

2 年前

话说Groupings这个函数你见过么.....那这个解起来特别方便

👍 赞

▲ 赞同 48



💬 10 条评论

➦ 分享

★ 收藏

👍 赞



wjxway

2 年前

```
With[{expr =  
  
HoldForm /@  
  
DeleteDuplicates@  
  
Groupings[  
  
Permutations[{a$, b$, c$, d$}, {4}], {Plus -> {2, Orderless},  
  
Subtract -> 2, Times -> {2, Orderless}, Divide -> 2}]],  
  
f[{a_, b_, c_, d_}] := Quiet@Select[expr, ReleaseHold[#] == 24 &]  
  
];
```

👍 赞



wjxway

2 年前

哇哦，成功在5s内算完仨数都在13以内的24点~

👍 赞



wjxway

2 年前

```
With[{  
  
expr =  
  
DeleteDuplicates@  
  
Groupings[  
  
Permutations[{a$, b$, c$, d$}, {4}], {Plus -> {2, Orderless},  
  
Subtract -> 2, Times -> {2, Orderless}, Divide -> 2}]], {  
  
exprh = HoldForm /@ expr},  
  
calc[{a_, b_, c_, d_}] := expr;  
  
calcp[{a_, b_, c_, d_}, i_] := exprh[[i]];
```

▲ 赞同 48



💬 10 条评论

➦ 分享

★ 收藏

知乎

首发于
次元壁

];

```
Block[{subs = Subsets[Range@13, {4}],
```

```
calcp[subs[[#1]], #2] & @@@
```

```
Quiet@Position[Transpose@calc@Transpose@subs,
```

```
24]]; // AbsoluteTiming
```

赞



wjxway

2 年前

smirk~

赞



wjxway

2 年前

啊.....不该用Subsets，假如用DeleteDuplicatesBy[Tuples[Range@13, 4], Sort]的话是10s~还是快很多哦~

赞



Kolkwitzia

2 年前

想起了中学是做的一堆死活算不出来的24点.....

赞

▲ 赞同 48



● 10 条评论

➦ 分享

★ 收藏