知乎 首发于 CODE Viens Vanité

代码: 最速输入问题

酱紫君
数学 话题的优秀回答者

已关注

编辑推荐

129 人赞同了该文章

屏幕上有1个初始字符, 只使用Ctrl+A, Ctrl+C, Ctrl+V, n个字符最少需要多少次按键?

其实就是整数分解

```
dp[i] = Min[dp[i], dp[j] + i/j + 2, dp[i/j] + j + 2]; dp[1] = 0
```

教一个小技巧, 匿函递归

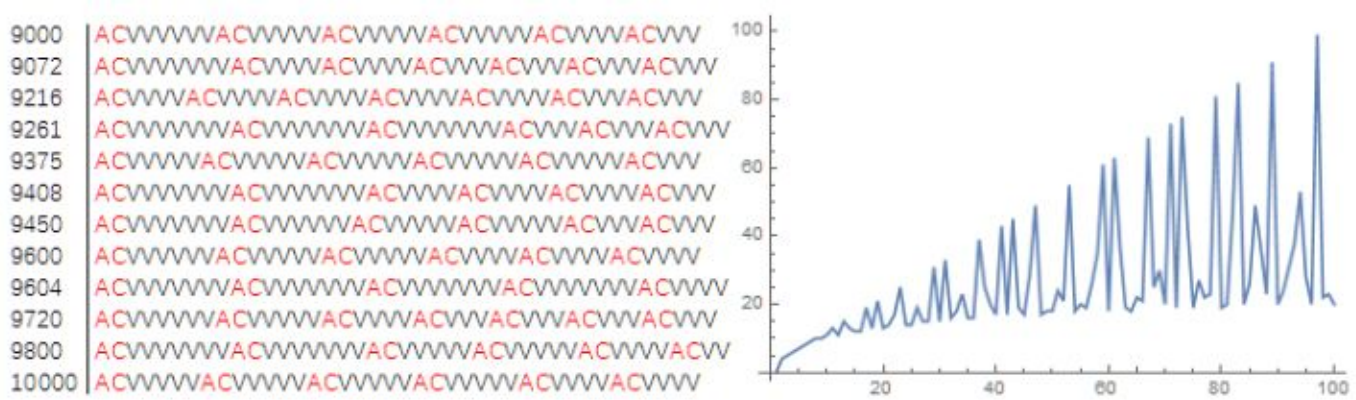
```
If[#==1,0,With[{l=Rest@Reverse@Divisors@#},Min[#0/@l+#/l+2]]]&[10000]
```

一个函数式语言都不能递归匿名函数那可能是对面混进来的奸细.....

如果要输出这个序列稍微麻烦点

```
stdp[i_]:=stdp[i]=Block[
        {l=Rest@Reverse@Divisors[i],st},
        st=Take[l,Ordering[dp/@l+i/l+2,1]];
        Join[stdp@@st,{"A","C"},ConstantArray["V",i/First@st]]
]
TableForm[StringJoin@@@stdp/@#,TableHeadings->{#,None}]&[DeleteCases[Range[30],
ListLinePlot[dp/@Range[100]]
```

算法复杂度是PrimeOmega函数, 和因子数量有关.



什么时候用粘贴键比较谜

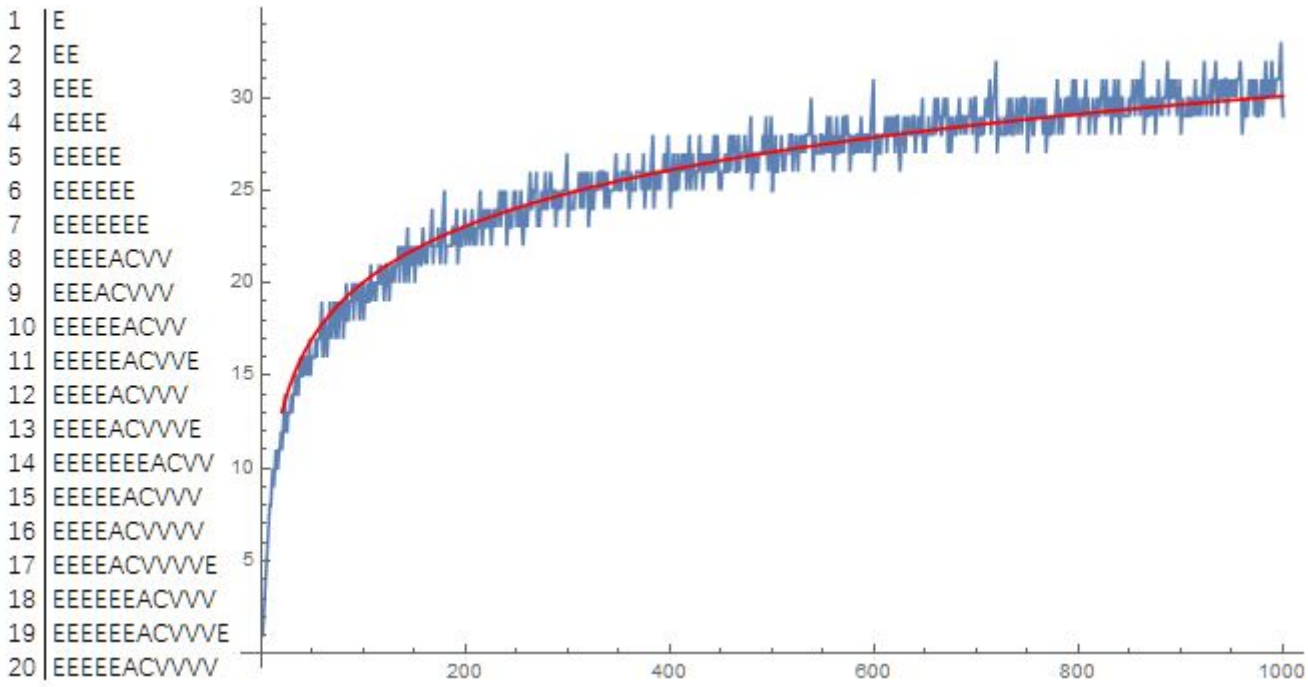
如果没有初始字符, 但是允许输入

这就比较有趣了

```
stdp[1]=ConstantArray["E",dp[1]=1];
dp[i_]:=dp[i]=With[
        {l=Rest@Reverse@Divisors[i]},
        Min@Append[dp/@l+i/l+2,dp[i-1]+1]
];
stdp[i_]:=stdp[i]=Block[
        {l=Rest@Reverse@Divisors[i],st},
        st=Ordering[Join[dp/@l+i/l+2,{dp[i-1]+1}],1];
        If[First@st>Length[l],
                Append[stdp[i-1],"
                Join[stdp@@Take[l,
```

```
TableForm[StringJoin@@@Array[stdp,20],TableHeadings->{Range[20],None}]
\[Alpha]=a/.FindFit[data=Array[dp,1000],a Log[x],a,x]
Show[ListLinePlot@data,Plot[a Log[x]/. {a->\[Alpha]},{x,1,1000},PlotStyle->Red]
```

但是因为要比较 **dp[i]** 和 **dp[i − 1]** ,复杂度一下子上去了.



按键次数符合对数曲线 $p(n) = 4.35639 \log n$, 数学原理不明...

mod["ParameterTable"]

	Estimate	Standard Error	t-Statistic	P-Value
a	4.35639	0.00490315	888.489	$5.09848676343 \times 10^{-1450}$

这个P值检验有毒...

再加入删除就比较难了,因为不满足最优子结构了

我们索性把难度推到顶

我们来把问题推广到最广义的情况

> 删除代价 D
> 全选代价 S
> 粘贴代价 V
> 注意第一次粘贴只会把当前内容覆盖掉.

@Lightwing 觉得可能要用抽象代数....好吧, 其实图论就够了...

翻n倍,全选+粘贴, 代价 **`S + nV`**

输入当然是从i-1输入, 代价 **`E`**

删除当然是从i+1删除, 代价 **`D`**

于是我们可以写出代价函数:

```
cost[i_]:=Block[
        {P=1,D=1,S=2,V=1,l},
        l=Drop[Divisors[i],-1];
        Join[
                Transpose[{Thread[l->i],S+i/l V}],
                {{i-1->i,P},{i+1->i,D}}
        ]
]
```

```
In[493]:= cost[24] // TableForm
Out[493]//TableForm=
1 → 24    26
2 → 24    14
3 → 24    10
4 → 24    8
6 → 24    6
8 → 24    5
12 → 24   4
23 → 24   1
25 → 24   1
```

于是以数字为节点, 转移关系为边, 代价为权, 规模小的时候(n<1000)我们可以使用图编程

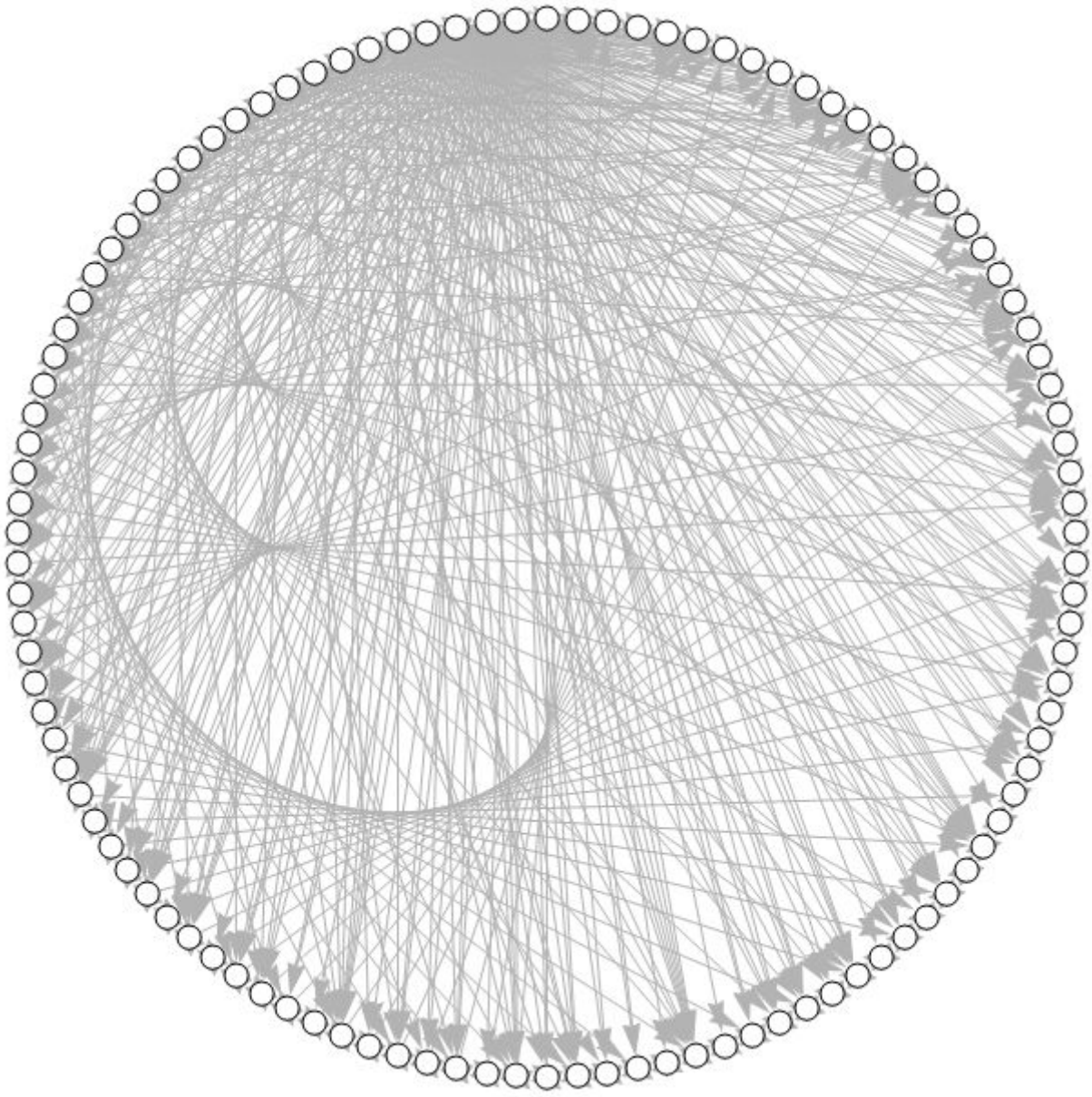
`Array[cost, 100]` 遍历整个求解域, 然后用 **`Graph`** 转化为图对象

但是Mathematica的图编程只能用在**单图**.

所以还要洗一下数据.

```
G=Graph[dir,EdgeWeight->wei,
        PlotTheme->"Monochrome",
        GraphLayout->"CircularEmbedding"
]
```

规模大...手动 **Indexed** 节点,然后 **BreadthFirstScan** 遍历吧...



左半面毫无疑问是分形结构...但是和整数分解相关...没啥好研究的

问题转变为求节点 a_0 到节点 a_n 的最短路

Dijkstra lalalaa ~~~ so easy...

```
        TableHeadings->{Range@20,None},
        TableAlignments->Left
]
MatrixForm[
        GraphDistanceMatrix[G][[1;;20,1;;20]]//Round,
        TableHeadings->{Range@20,Range@20}
]
```

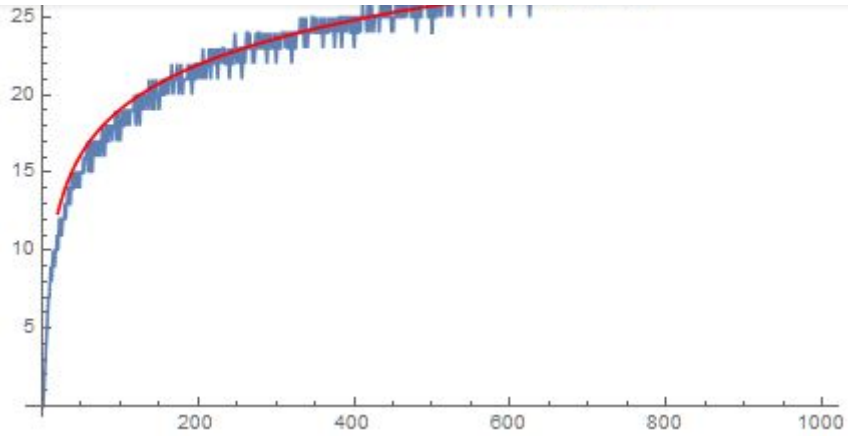
内置的 **GraphDistanceMatrix** 只能用于单图, 如果你装了IGraphM扩展包的话 **IGraphM`IGDistanceMatrix** 可以用于多重图.

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

最左边一列1没单独处理,所以出错了,其实就是 j - i

输入和删除代价调到无穷大...等会儿, 无穷大会出错...调到10000就行, 退化到第一种

单单 删除代价调高会慢慢退化到第二种情况.



虽然看上去没有变化但实际上下降到了 $p(n) = 4.1414 \log n$ ，数学原理同样不明

代价仍然不是单增的...这个评论就很想当然了.....

编辑于 2018-01-10

Wolfram Mathematica 数学 算法

知乎 首发于 CODE Viens Vanité

却在不断传播。这个问题已经传遍

中科院物理所

金重

成语接龙

17 条评论

切换为时间排序

写下你的评论...

QwQwQwQ 2 年前

居然把这个做出来了😂

2

WXSB 回复 **QwQwQwQ** 2 年前

大佬 电脑什么配置啊，

赞

于晨阳 2 年前

稳

1

于晨阳 2 年前

那个图让我想起了hatcher的封面。。

赞

SulphurForKep 2 年前

求封面图来源...

赞

月明星稀 2 年前

我靠

赞

wind void 2 年前

棒啊！

1

triple six

赞

王赟 Maigo 2 年前

分形图好像一只耳朵

赞

大铀子 2 年前

这题目包括鼠标按键吗

赞

酱紫君 (作者) 回复 大铀子 2 年前

那题不是被人干了吗

赞

大铀子 回复 酱紫君 (作者) 2 年前

我没看过题

赞

MedalPluS 2 年前

是我读错了还是啥 为啥需要dp这题?

赞

有道之士 2 年前

这个问题改一改：允许n次输入，求最长输出，就单增了。

仔细想想，难度上不是一个次元了~

赞

酱紫君 (作者) 回复 有道之士 2 年前

这个问题超简单啊...甚至有公式解

1

有道之士 回复 酱紫君 (作者) 2 年前

是啊，简单到跟原题不是一个次元了~

赞

何从