

什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



算法 数学 计算机科学 运筹学 动态规划

关注者
15,478

被浏览
2,697,192

什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什么?

关注问题

写回答

邀请回答

好问题 418

13 条评论

分享 ...

查看全部 217 个回答



阮行止

网络安全研究者 | 前算法竞赛选手

+ 关注

专业 已有 5 人赠与了专业徽章 >

15,096 人赞同了该回答

0. intro

很有意思的问题。以往见过许多教材，对动态规划 (DP) 的引入属于“奉天承运，皇帝诏曰”式：不给出一点引入，见面即拿出一大堆公式吓人；学生则死啃书本，然后突然顿悟。针对入门者的教材不应该是这样的。恰好我给入门者讲过四次DP入门，迭代出了一套比较靠谱的教学方法，所以今天跑过来献丑。

现在，我们试着自己来一步步“重新发明”DP。

1. 从一个生活问题谈起

先来看看生活中经常遇到的事吧——假设您是个土豪，身上带了足够的1、5、10、20、50、100面值的钞票。现在您的目标是凑出某个金额w，需要用到尽量少的钞票。

依据生活经验，我们显然可以采取这样的策略：能用100的就尽量用100的，否则尽量用50的.....依次类推。在这种策略下， $666=6\times 100+1\times 50+1\times 10+1\times 5+1\times 1$ ，共使用了10张钞票。

这种策略称为“贪心”：假设我们面对的局面是“需要凑出w”，贪心策略会尽快让w变得更小。能让w少100就尽量让它少100，这样我们接下来面对的局面就是凑出w-100。长期的生活经验表明，贪心策略^Q是正确的。

但是，如果我们换一组钞票的面值，贪心策略也许不成立了。如果一个奇葩国家的钞票面额分别是1、5、11，那么我们在凑出15的时候，贪心策略会出错：

$15=1\times 11+4\times 1$ (贪心策略使用了5张钞票)

$15=3\times 5$ (正确的策略，只用3张钞票)

为什么会这样呢？贪心策略错在了哪里？

鼠目寸光。

刚刚已经说过，贪心策略的纲领是：“尽量使接下来面对的w更小”。这样，贪心策略在w=15的局面时，会优先使用11来把w降到4；但是在这个问题中，凑出4的代价是很高的，必须使用4x1。如果使用了5，w会降为10，虽然没有4那么小，但是凑出10只需要两张5元。

在这里我们发现，贪心是一种只考虑眼前情况的策略。

那么，现在我们怎样才能避免鼠目寸光呢？

如果直接暴力枚举凑出w的方案，明显复杂度过高。太多方法可以凑出w了，枚举它们的时间是不可承受的。我们现在来尝试找一下性质。



关于作者



阮行止

网络安全研究者 | 前算法竞...

回答

2

文章

1

关注者

10,386

+ 关注他

发私信

被收藏 12,118 次

赞同超3k的回答-第一辑 39,571 人关注
江湖人称向前兄 创建

有趣的知识 17,553 人关注
天猪 创建

奇奇怪怪的东西 1,993 人关注
刘镇锐 创建

程序开发 1,319 人关注
梦回琼华 创建

优秀的经济学科普文 891 人关注
刘镇锐 创建

相关问题

如何才能写好动态规划 (Dynamic Programming) ? 0 个回答

该如何解释动态规划? 0 个回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

张? “接下来, 我们用 $f(n)$ 来表示“凑出 n 所需的最少钞票数量”。

那么, 如果我们取了11, 最后的代价 (用掉的钞票总数) 是多少呢?

明显 $\text{cost} = f(4) + 1 = 4 + 1 = 5$, 它的意义是: 利用11来凑出15, 付出的代价等于 $f(4)$ 加上自己这一张钞票。现在我们暂时不管 $f(4)$ 怎么求出来。

依次类推, 马上可以知道: 如果我们用5来凑出15, cost 就是 $f(10) + 1 = 2 + 1 = 3$ 。

那么, 现在 $w=15$ 的时候, 我们该取那种钞票呢? 当然是各种方案中, cost 值最低的那一个!

- 取11: $\text{cost} = f(4) + 1 = 4 + 1 = 5$
- 取5: $\text{cost} = f(10) + 1 = 2 + 1 = 3$
- 取1: $\text{cost} = f(14) + 1 = 4 + 1 = 5$

显而易见, cost 值最低的是取5的方案。我们通过上面三个式子, 做出了正确的决策!

这给了我们一个至关重要的启示—— $f(n)$ 只与 $f(n-1), f(n-5), f(n-11)$ 相关; 更确切地说:

$$f(n) = \min\{f(n-1), f(n-5), f(n-11)\} + 1$$

这个式子是非常激动人心的。我们要求出 $f(n)$, 只要求出几个更小的 f 值; 既然如此, 我们从小到大把所有的 $f(i)$ 求出来不就好了? 注意一下边界情况即可。代码如下:

```
int f[105], i, n, cost;
scanf("%d", &n);

f[0] = 0;

for (i = 1; i <= n; i++)
{
    cost = INF;
    if (i-1 >= 0) cost = min(cost, f[i-1]+1);
    if (i-5 >= 0) cost = min(cost, f[i-5]+1);
    if (i-11 >= 0) cost = min(cost, f[i-11]+1);
    f[i] = cost;
    printf("f[%d] = %d\n", i, f[i]);
}
```

```
C:\Users\Administrator\Desktop\讲\code\coin.exe
15
f[1]=1
f[2]=2
f[3]=3
f[4]=4
f[5]=1
f[6]=2
f[7]=3
f[8]=4
f[9]=5
f[10]=2
f[11]=1
f[12]=2
f[13]=3
f[14]=4
f[15]=3
3
```

我们以 $O(n)$ 的复杂度解决了这个问题。现在回过头来, 我们看看它的原理:

- $f(n)$ 只与 $f(n-1), f(n-5), f(n-11)$ 的值相关。
- 我们只关心 $f(w)$ 的值, 不关心是怎么凑出 w 的。

这两个事实, 保证了我们做法的正确性。它比起贪心策略, 会分别算出取1、5、11的代价, 从而做出一个正确决策, 这样就避免掉了“鼠目寸光”!

它与暴力的区别在哪里? 我们的暴力枚举了“使用的硬币”, 然而这属于冗余信息。我们要的是答案, 根本不关心这个答案是怎么凑出来的。譬如, 要求出 $f(15)$, 只需要知道 $f(14), f(10), f(4)$ 的值。其他信息并不需要。我们舍弃了冗余信息。我们只记录了对解决问题有帮助的信息—— $f(n)$ 。

我们能这样干, 取决于问题的性质: 求出 $f(n)$, 只需要知道几个更小的 $f(c)$ 。我们将求解 $f(c)$ 称作求解 $f(n)$ 的“子问题”。

这就是DP (动态规划, dynamic programming)。

将一个问题拆成几个子问题, 分别求解这些子问题, 即可推断出大问题的解。

思考题: 请稍微修改代码, 输出我们凑出 w 的方案。

[关注问题](#)[写回答](#)

如何理解动态规划的意义是什么? 0 个回答

动态规划和贪心算法的异同, 如何区分? 6 个回答

相关推荐



人体运动数据处理及合成中的关键算法

5 人读过

阅读



概率的概率

acel rovsion

36,991 人读过

阅读



计算机数学: 算法基础 线性代数与图论

48 人读过

阅读

刘看山 · 知乎指南 · 知乎协议 · 知乎隐私保护指引

应用 · 工作 · 申请开通知乎机构号

侵权举报 · 网上有害信息举报专区

京 ICP 证 110745 号

京 ICP 备 13052560 号 - 1

京公网安备 11010802020088 号

京网文[2022]2674-081 号

药品医疗器械网络信息服务备案

(京) 网药械信息备字 (2022) 第00334号

服务热线: 400-919-0001

违法和不良信息举报: 010-82716601

举报邮箱: jubao@zhihu.com

儿童色情信息举报专区

互联网算法推荐举报专区

养老诈骗举报专区

MCN 举报专区

信息安全漏洞反馈专区

内容从业人员违法违规举报

网络谣言信息举报入口

证照中心 · Investor Relations

联系我们 © 2023 知乎

北京智者天下科技有限公司版权所有



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

[关注问题](#)[写回答](#)

【无后效性^Q】

一旦 $f(n)$ 确定, “我们如何凑出 $f(n)$ ”就再也用不着了。

要求出 $f(15)$, 只需要知道 $f(14), f(10), f(4)$ 的值, 而 $f(14), f(10), f(4)$ 是如何算出来的, 对之后的问题没有影响。

“未来与过去无关”, 这就是无后效性。

(严格定义: 如果给定某一阶段的状态, 则在这一阶段以后过程的发展不受这阶段以前各段状态的影响。)

【最优子结构^Q】

回顾我们对 $f(n)$ 的定义: 我们记“凑出 n 所需的最少钞票数量”为 $f(n)$ 。

$f(n)$ 的定义就已经蕴含了“最优”。利用 $w=14, 10, 4$ 的最优解, 我们即可算出 $w=15$ 的最优解。

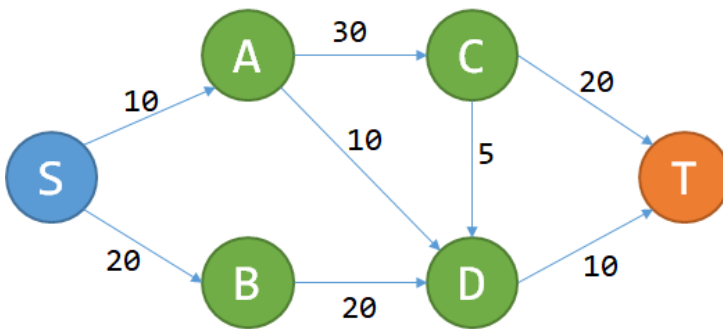
大问题的最优解可以由小问题的最优解推出, 这个性质叫做“最优子结构性^Q”。

引入这两个概念之后, 我们如何判断一个问题能否使用DP解决呢?

能将大问题拆成几个小问题, 且满足无后效性、最优子结构性。

3. DP的典型应用: DAG最短路^Q

问题很简单: 给定一个城市的地图, 所有的道路都是单行道, 而且不会构成环。每条道路都有过路费, 问您从S点到T点花费的最少费用。



一张地图。边上的数字表示过路费。

这个问题能用DP解决吗? 我们先试着记从S到P的最少费用为 $f(P)$ 。

想要到T, 要么经过C, 要么经过D。从而 $f(T) = \min \{f(C) + 20, f(D) + 10\}$ 。

好像看起来可以DP。现在我们检验刚刚那两个性质:

- 无后效性: 对于点P, 一旦 $f(P)$ 确定, 以后就只关心 $f(P)$ 的值, 不关心怎么去的。

- 最优子结构: 对于P, 我们当然只关心到P的最小费用, 即 $f(P)$ 。如果我们从S走到T是 $S \rightarrow P \rightarrow Q \rightarrow T$, 那肯定S走到Q的最优路径是 $S \rightarrow P \rightarrow Q$ 。对一条最优的路径而言, 从S走到沿途上所有的点(子问题)的最优路径, 都是这条大路的一部分。这个问题的最优子结构性是显然的。

既然这两个性质都满足, 那么本题可以DP。式子明显为:



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

[关注问题](#)[写回答](#)

其中 R 为有路通到 P 的所有的点, $w_{R \rightarrow P}$ 为 R 到 P 的过路费。

代码实现也很简单, 拓扑排序即可。

4. 对DP原理的一点讨论

【DP的核心思想】

DP为什么会快?

无论是DP还是暴力, 我们的算法都是在可能解空间内, 寻找最优解。

来看钞票问题。暴力做法是枚举所有的可能解, 这是最大的可能解空间。

DP是枚举有希望成为答案的解。这个空间比暴力的小得多。

也就是说: DP自带剪枝。

DP舍弃了一大堆不可能成为最优解的答案。譬如:

$15 = 5+5+5$ 被考虑了。

$15 = 5+5+1+1+1+1+1$ 从来没有考虑过, 因为这不可能成为最优解。

从而我们可以得到DP的核心思想: 尽量缩小可能解空间。

在暴力算法中, 可能解空间往往是指数级的大小; 如果我们采用DP, 那么有可能把解空间的大小降到多项式级⁹。

一般来说, 解空间越小, 寻找解就越快。这样就完成了优化。

【DP的操作过程】

一言以蔽之: 大事化小, 小事化了。

将一个大问题转化成几个小问题;

求解小问题;

推出大问题的解。

【如何设计DP算法】

下面介绍比较通用的设计DP算法的步骤。

首先, 把我们面对的局面表示为 x 。这一步称为设计状态。

对于状态 x , 记我们要求出的答案(e.g. 最小费用)为 $f(x)$. 我们的目标是求出 $f(T)$.

找出 $f(x)$ 与哪些局面有关 (记为 p), 写出一个式子 (称为状态转移方程), 通过 $f(p)$ 来推出 $f(x)$.

【DP三连】

设计DP算法, 往往可以遵循DP三连:

我是谁? ——设计状态, 表示局面

我从哪里来?

我要到哪里去? ——设计转移

设计状态是DP的基础。接下来的设计转移, 有两种方式: 一种是考虑我从哪里来 (本文之前提到的两个例子, 都是在考虑“我从哪里来”); 另一种是考虑我到哪里去, 这常见于求出 $f(x)$ 之后, 更新能从 x 走到的一些解。这种DP也是不少的, 我们以后会遇到。

总而言之, “我从哪里来”和“我要到哪里去”只需要考虑清楚其中一个, 就能设计出状态转移方



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

[关注问题](#)[写回答](#)

思考题：如何把钞票问题的代码改写成“我到哪里去”的形式？

提示：求出 $f(x)$ 之后，更新 $f(x+1), f(x+5), f(x+11)$ 。

5. 例题：最长上升子序列^Q

扯了这么多形而上的内容，还是做一道例题吧。

最长上升子序列 (LIS) 问题：给定长度为 n 的序列 a ，从 a 中抽取出一个子序列，这个子序列需要单调递增。问最长的上升子序列 (LIS) 的长度。

e.g. 1,5,3,4,6,9,7,8的LIS为1,3,4,6,7,8，长度为6。

如何设计状态（我是谁）？

我们记 $f(x)$ 为以 a_x 结尾的LIS长度，那么答案就是 $\max\{f(x)\}$ 。

状态 x 从哪里推过来（我从哪里来）？

考虑比 x 小的每一个 p ：如果 $a_x > a_p$ ，那么 $f(x)$ 可以取 $f(p)+1$ ^Q。

解释：我们把 a_x 接在 a_p 的后面，肯定能构造一个以 a_x 结尾的上升子序列，长度比以 a_p 结尾的LIS大1。那么，我们可以写出状态转移方程了：

$$f(x) = \max_{p < x, a_p < a_x} \{f(p)\} + 1$$

至此解决问题。两层for循环，复杂度 $O(n^2)$ 。

```
int main(void)
{
    int f[105]={0},a[105]={0},i,x,p,n,ans=0;
    scanf("%d",&n);
    for(i=1;i<=n;i++)
        scanf("%d",&a[i]),f[i]=1;

    for(x=1;x<=n;x++)
    {
        for(p=1;p<x;p++)
            if(a[p]<a[x]) f[x]=max(f[x],f[p]+1);
        printf("f[%d]=%d\n",x,f[x]);
    }

    for(x=1;x<=n;x++)
        ans=max(ans,f[x]);

    printf("%d\n",ans);
}
```

```
C:\Users\Administrator\Desktop\讲课\code\lcs.
8
1 5 3 4 6 9 7 8
f[1]=1
f[2]=2
f[3]=2
f[4]=3
f[5]=4
f[6]=5
f[7]=5
f[8]=6
6
-----
Process exited after 4.065 seconds w
请按任意键继续. . .
```

从这三个例题中可以看出，DP是一种思想，一种“大事化小，小事化了”的思想。带着这种思想，DP将会成为我们解决问题的利器。

最后，我们一起念一遍DP三连吧——我是谁？我从哪里来？我要到哪里去？

6. 习题

如果读者有兴趣，可以试着完成下面几个习题：

一、请采取一些优化手段，以 $O(n \log n)$ 的复杂度解决LIS问题。

提示：可以参考这篇博客 [Junior Dynamic Programming--动态规划初步·各种子序列问题](#)



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



三、01背包问题是一种常见的DP模型。请完成P1048 采药 - 洛谷。

感谢您看完本文 /Q//w//Q/

2019.3.3

编辑于 2019-03-03 17:09

赞同 1.5 万



352 条评论

分享

收藏

喜欢



收起 ^

更多回答



苗华栋



大连理工大学 控制理论与控制工程硕士

+ 关注

1,620 人赞同了该回答

希望本文不仅能告诉你什么是动态规划，也能给你一种如何分析、求解动态规划问题的思考方式。

0001b 动态规划介绍

1. 运筹学中的动态规划

动态规划(Dynamic Programming, 简称DP)是运筹学的一个分支。它是解决多阶段决策过程最优化的一种数学方法。把多阶段问题变换为一系列相互联系的单阶段问题，然后逐个加以解决。

这里提到动态规划其实是一种数学方法，是求解某类问题的一种方法，而不是一种特殊的算法，没有一个标准的数学表达式或明确定义的一种规则。

比如我们接触过的“排序算法”，“二叉树遍历算法”等，这些算法都是有固定范式的。遇到此类问题我们只需要照搬算法即可。但动态规划却不是，它告诉你的是解决某类问题的一种思路，或者是一种更高意义上的算法。是一种道，而不是术。所以动态规划难就难在我们即使学会了这种思想，遇到具体问题也需要具体分析，很可能因为我们构造不出动态规划所需要的形式而无法解决。甚至根

[展开阅读全文](#)

赞同 1620



20 条评论

分享

收藏

喜欢



帅地



+ 关注

2,747 人赞同了该回答

之前写过一篇入门动态规划地文章。文章中给出了动态规划地套路和优化方案。在知乎收获了不少赞。也收到挺多人感谢

00:47



评论与转发

最新 · 7



Ennio 评论了你的文章

3 小时前

[展开阅读全文](#)



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

 写回答



查看全部 217 个回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

 写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



关注问题

 写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



什么是动态规划 (Dynamic Programming) ? 动态规划的意义是什...

关注问题

写回答



关注问题

 写回答

