

昵称：Geek_Ling
园龄：5年8个月
粉丝：335
关注：9
+加关注

< 2011年11月 >

日	一	二	三	四	五	六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

搜索

找找看

谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签
更多链接

最新随笔

1. The Triumph Of Bio-logic
2. 从程序员到猎头
3. 那些年看过的书
4. Linux 文件操作总结
5. 说说计算机中的异常
6. linux系统调用和库函数调用的区别
7. 词法分析器的实现
8. 对快速排序的简单分析
9. C中的循环是如何实现的
10. 从hello world 说程序运行机制

我的标签

操作系统(4)
算法(4)
递归(3)
数组(2)
死锁(2)
指针(2)
动态规划(2)
动态规划实例(2)
内存(2)
系统调用(2)
更多

随笔分类(35)

.NET学习(1)
《失控》(1)
【操作系统相关】operating systems(8)
C++(2)
Linux(1)
SQL技术(1)
编程之美(2)
编译原理(1)
程序性能优化(4)
大数据
计算机网络(computer networking)

动态规划解——有向图中的最长路径

动态规划博大精深，想完全掌握是很难的，不过我们可以从一些简单的例子之中去体会她的奥妙。

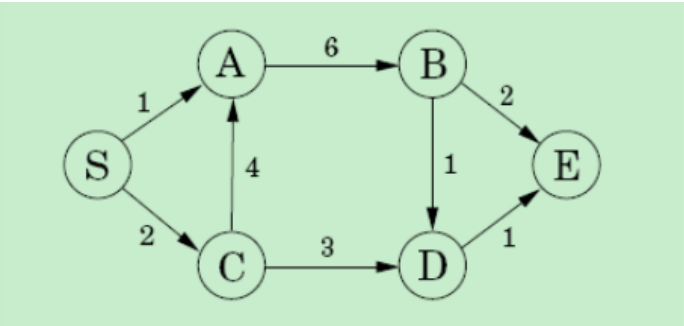
不说废话、先来一个简单的例子吧：

longest path in DAG

Problem:
Given a weighted directed acyclic graph $G=(V, E)$, an vertex v , where each edge is assigned an integer weight, find a longest path in graph G

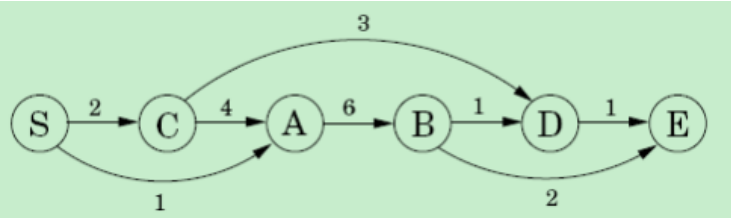
问题描述：

给一个带权有向无环图 $G=(V, E)$ ，找出这个图里的最长路径。



说实话初学者直接给出这个图会看蒙的、再看看问题，不知道从何下手。

好了，对上图做个简单的处理：



现在看起来是不是清晰多了呢

用 $dilg(v)$ 表示 以点结尾的最长路径，现在考虑 $dilg(D)$, $dilg(B)$, $dilg(C)$

$$dilg(D)=\max\{dilg(B)+1, dilg(C)+3\}$$

来解释一下：点D的入度边有CD、BD。

以D结尾的最短路径必定经过C、D中的最后一点；如果是C点，则以 $dilg(C)+3$ (权值)定会大于等于 $dilg(D)+2$ (权值)

如果没能看懂，请注意 $dilg(V)$ 的定义

对于任意的点V可以有如下表达式：

$$dilg(v)=\max_{(u,v)\in E}\{dilg(u)+w(u, v)\}$$

这样、问题 $dilg(V)$ 就会被转化为较小的子问题 $dilg(U)$ （当然，U是连入V的点）

任何一个问题都可以用这样的方式转化、变小。

但总不能无限变小啊，最后回成为最简单的问题。当有两个点，且其中的一个点入度为0的时候如图中的S ->C他们的最长距离就是

权值2。入门篇中说过，思考方向是从复杂到简单，而计算方向是从简单到复杂。

算法如下

算法和数据结构(8)
我的翻译(1)
知其所以然(5)

随笔档案(37)

2014年12月 (1)
2014年7月 (1)
2013年11月 (1)
2012年8月 (1)
2012年5月 (1)
2012年4月 (3)
2012年3月 (2)
2012年2月 (7)
2012年1月 (2)
2011年12月 (7)
2011年11月 (11)

文章分类

SQL技术

积分与排名

积分 - 96706
排名 - 2762

最新评论

- Re:C++ 值传递、指针传递、引用传递详解
谢谢
--漫路求索者
- Re:词法分析器的实现
很好的一篇文章
--jakefeng
- Re:Hanoi Tower 汉诺塔的简单分析/C #-* -encoding:utf-8-* - #递归就是减少问题的规模def hanoi(n, a, b, c): if n == 1: move(n, a, c) #一个的时候a移到.....
--焚心似火
- Re:Hanoi Tower 汉诺塔的简单分析/C 对递归有了一个更好的认识,但是状态2应该是由b移到c吧?
--suwenyuan
- Re:Hanoi Tower 汉诺塔的简单分析/C 显然,只要把B上的n-1个盘子移动到A,待解决的问题和原问题就相比就只是规模变小了。这里面的n-1觉得应该是n-2。因为要将B上的n-2个盘子移动到A,然后剩下B的第n-2盘子移动到C...
--河马不懂英语

阅读排行榜

- 词法分析器的实现(74988)
- C++ 值传递、指针传递、引用传递详解(53432)
- Linux 文件操作总结(46199)
- Hanoi Tower 汉诺塔的简单分析/C(21271)
- 对缓存的思考——提高命中率(20899)

评论排行榜

- Hanoi Tower 汉诺塔的简单分析/C(27)
- 说说哪本书是对程序员最有影响、每个程序员都该阅读的书?(20)
- 对缓存的思考——提高命中率(17)
- 词法分析器的实现(16)
- 我认识的线程(11)

推荐排行榜

- 对缓存的思考——提高命中率(25)
- 词法分析器的实现(15)
- C++ 值传递、指针传递、引用传递详解(13)
- Hanoi Tower 汉诺塔的简单分析/C(12)
- 我认识的线程(10)

Initialize all $dilg(.)$ values to ∞ ;

1. Let S be the set of vertices with indegree=0;
的点

////设集合S, 里面的入度为0

2. For each vertex v in S do
 $dilg(v)=0$;

3. For each $v \in V \setminus S$ in Topological Sorting order do
顺序, 依次求出最长路径并保存好

//对于V中除S外的点、按照拓扑排序的

$dilg(v)=\max_{(u,v) \in E} \{dilg(u)+w(u, v)\}$
到终点的最短路径

//拓扑排序可以简单的理解为从起点

4. Return the $dilg(.)$ with maximum value.

现在是找到最长路径的大小了、但是如何得到最长路径呢?
只需做点小小的改动:

Dplongestpath(G)

Initialize all $dilg(.)$ values to ∞ ;

Let S be the set of vertices with indegree=0;
for each vertex v in S do

$dist(v)=0$;
4. For each $v \in V \setminus S$ in Topological Sorting order do
 $dilg(v)=\max_{(u,v) \in E} \{dilg(u)+w(u, v)\}$
let (u,v) be the edge to get the maximum value;
 $dad(v)=u$;

5. Return the $dilg(.)$ with maximum value.

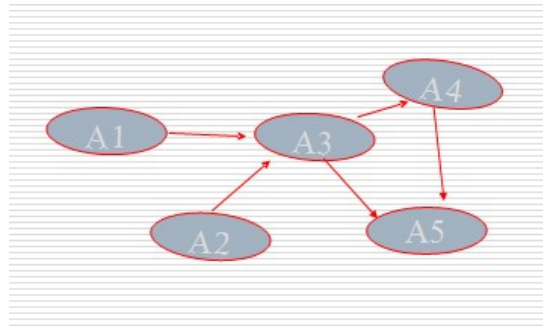
每一步都记下V的父节点、最后根据dad()数组即可得到路径。

对于上面的问题: 先找出子问题、然后解决由子问题如何得到父问题以及如何把子问题分为更小的子问题

注意: 问题的本质没有改变, 只是规模变小。

我的理解:

动态规划的目的正是在不改变问题本质的情况下不断缩小子问题的规模、规模很小的时候, 也就能很容易得到解啦(正如上面的只有两个点的情况)



上图可以这样理解:

问题A5被分成了子问题A3、A4解决它们就能解决A5, 由于A3、A4和A5有相同的结构(问题的本质没变)所以A3可以分为问题A1、A2。当然A4也能

分为两个子问题, 只是图中没画出来。

下面的是再网上看到的不错的思路:

Dynamic programming:

- (1) problem is solved by identifying a collection of subproblems,
- (2) tackling them one by one, smallest rst,
- (3) using the answers of small problems to help figure out larger ones,
- (4) until the whole lot of them is solved.

第一次觉得要讲清楚一件事还挺不容易的O(∩_∩)O~大半个晚上才搞定得。。。汗!

有不得当的地方希望高手指正呀!

分类: [算法和数据结构](#)

标签: [dynamic programming](#), [动态规划实例](#), [Longest path in DAG](#), [最长路径问题](#)

好文要顶

关注我

收藏该文

Geek_Ling

关注 - 9

粉丝 - 335

+加关注

4

1

« 上一篇 : [dynamic programming动态规划初步理解【-1】](#)
» 下一篇 : [Hanoi Tower 汉诺塔的简单分析/C](#)

posted @ 2011-11-12 20:28 Geek_Ling 阅读(16733) 评论(5) 编辑 收藏

发表评论

#1楼 2011-11-12 22:12 | hello GG

以D结尾的最短路径必定经过C、D中的最后一点；如果是C点，则以dilg(C)+3(权值)定会大于等于dilg(D)+2(权值)

不明白，为什么是以D结尾的最短路径，开始不是将最长路径吗？

支持(0) 反对(0)

#2楼[楼主] 2011-11-12 22:48 | Geek_Ling

@ hello GG

非常抱歉的细节上确实有错误的地方、晕、、打字错了刚刚看的时候确实有地方没讲清楚的、、dilg(V)的定义为：以点结尾的最长路径；

要构成以D结尾的路径，只能经过C、D两点；

因为D的入度边上相连的只有C、D两个点，所以以D点结尾的路径就一定会经过C、D点之一；

如果最长路径是经过的是C点，那么dilg(C)加上边CD的权值3就大于dilg(B)+权值1；

反之如是D点则应该是小于。

所以才会有

$$dilg(v)=\max(u,v)\in E\{dilg(u)+w(u,v)\}$$

其中U是和V的入度边相连的点、W(U,V)是边U，V的权值

选出最大的、就是路径中V的前驱点。

支持(0) 反对(0)

#3楼 2011-11-13 13:22 | hello GG

@ 一条鱼~

谢谢楼主的回复，您讲的很清楚了。应该不小心打错字了

D的入度相连的是B、C点，所以D的最长路径一定经过B、C两点之一

支持(0) 反对(0)

#4楼[楼主] 2011-11-13 13:36 | Geek_Ling

也谢谢你呀~其实

这只是一个很简单的例子、后面还会在比较经典的背包问题(knapsack)

、all pairs shortest in path\还有著名的旅行商问题、、

我想把动态规划的几个入门级例子都尽量写下来、

还有、真不好意思、以后会避免打错字这种情况啦、、

支持(2) 反对(0)

#5楼 2015-01-05 13:30 | Mr_sqw

楼主萌萌哒

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
【免费】从零开始学编程，开发者专属实验平台免费实践！

【推荐】现在注册又拍云，首月可享 200G CDN流量，还可免费申请 SSL 证书

【推荐】阿里云“全民云计算”优惠升级



最新IT新闻:

- 微软开始在Photos应用程序中推出AI人工智能图像搜索
 - 易到司机提现延期背后：或因乐视想用锅14亿债务遭拒
 - 苹果林业计划收获100%的纸张耗材供应
 - 夏普连续三个季度盈利 富士康削减成本措施奏效
 - 中兴上半年净利同比大增近30% 内部合规治理初见成效
- » 更多新闻...



最新知识库文章:

- 小printf的故事：什么是真正的程序员？
 - 程序员的工作、学习与绩效
 - 软件开发为什么很难
 - 唱吧DevOps的落地，微服务CI/CD的范本技术解读
 - 程序员，如何从平庸走向理想？
- » 更多知识库文章...

Copyright ©2017 Geek_Ling