Algorithm 0x01

49.234.77.58/index.php/2019/12/15/algorithm-0x01

XZLang 2019年12月15日

动态规划

- 可以使用动态规划的问题特点
  - 一个大问题可以划分为若干个子问题

    子问题：性质一样但是规模变小的问题

    最优子结构
  - 在计算时子问题有交叠。（交叠子结构）
- 记忆化思想

  对较小的子问题进行一次求解，并把结果记录下来，然后利用较小问题的解，求解出较大问题的解，直到求解出最大问题的解。[同一件事情不做第二次]

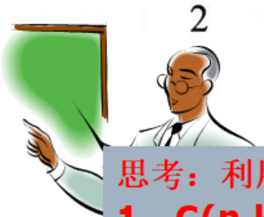
二项式系数算法

算法 Binomial(n,k)

```
//用动态规划算法计算C(n,k)
//输入：一对非负整数n≥k ≥0
//输出：C(n,k)的值
for(i←0;i<=n;i++)
   for(j←0;j<=min(i,k);j++)
      if(j=0 || j=k)   C[i,j]=1;
      else
          C[i,j]←C[i-1,j-1]+C[i-1,j];
Return C[n,k];
```

$$A(n,k)=\sum_{i=1}^{k}\sum_{j=1}^{i-1}1+\sum_{i=k+1}^{n}\sum_{j=1}^{k}1=\sum_{i=1}^{k}(i-1)+\sum_{i=k+1}^{n}k$$

$$=\frac{(k-1)k}{2}+k(n-k)\in\Theta(nk)$$



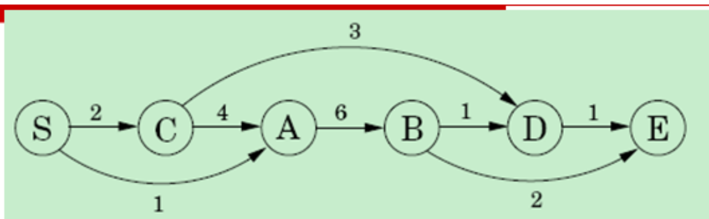
思考：利用公式

1、C(n,k)=n!/k!(n-k)!

2、C(n,k)=n(n-1)...(n-k+1)/k!

与动态规划计算比较，那一个的效率高？

Longest path in DAG



dilg(v): the longest path ending with v.

dilg(D), dilg(B), dilg(C)

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

For any vertex v,

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

Edit Distance

编辑距离（Minimum Edit Distance，MED），由俄罗斯科学家 Vladimir Levenshtein 在1965年提出，也因此而得名 Levenshtein Distance。

在信息论、语言学和计算机科学领域，Levenshtein Distance 是用来度量两个序列相似程度的指标。通俗地来讲，编辑距离指的是在两个单词 $< w_1, w_2 >$ 之间，由其中一个单词 w_1 转换为另一个单词 w_2 所需要的最少**单字符编辑操作**次数。

Knapsack

0-1 **背包，取第**j**个：**K(w-wj,j-1)+vj
不取第j**个：**K(w,j-1)

Chain matrix multiplication

Dpmatrixmul(A1, A2, .., An)

1. For i=1 to n
   C(i, i)=0;
2. For s=1 to n-1 do
   for i=1 to n-s do
     j=i+s;
     $C(i, j)=\min\{C(i,k)+C(k+1, j)+m_{i-1}.m_k.m_j,\ i\le k<j\}$;
3. Return C(1, n).

Running time: $O(n^3)$

Multiply four matrices: $A\times B\times C\times D$

A: 50×20 B: 20×1 C: 1×10 D: 10×100

All pairs shortest path

Dpallpairst(G)

1. For i=1 to n do
   for j=1 to n do
     $d_{ij}^{(0)}=\infty$;
2. For all $(i,j)\in E$ do
   $d_{ij}^{(0)}=w(i, j)$;
3. For k=1 to n do
   for i=1 to n do
     for j=1 to n do
       $d_{ij}^{(k)}=\min\{d_{ij}^{(k-1)},\ d_{ik}^{(k-1)}+d_{kj}^{(k-1)}\}$
4. Return $D^{(n)}$.

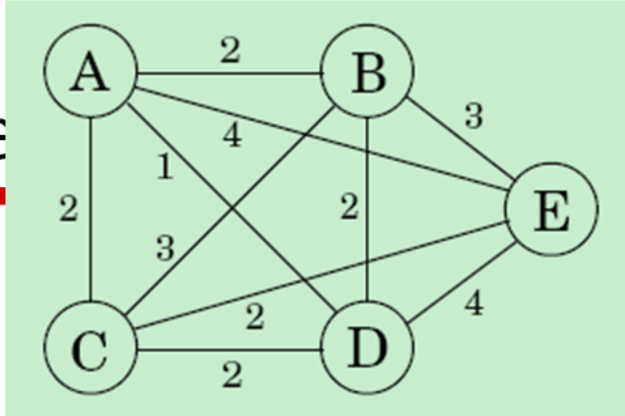
Running time: $O(n^3)$

弗洛伊德算法

Traveling Salesman

给定一个加权完全图G=（V，E），其中每一条边都有一个权重，找到一个简单的循环正好通过每个顶点一次，总权重最小化。

Traveling Sale



DPTSP(G)

1. $C(\{1\}, 1)=0$;
2. for $h=2$ to n do

    for all subsets $S \subseteq \{1,2,\ldots n\}$ of size h and containing 1 do

    $C(S,1)= \infty$;

    for all $j \in S$, $j \neq 1$ do

    $C(S, j)=\min\{C(S-\{j\}, i)+d_{ij}, i \in S, i \neq j\}$;

3. Return $\min_j C(\{1,2,..n\}, j)+d_{j1}$.

动态规划的一般步骤

- 最优解的结构特征
- 找到子问题的形式
- 用更小的子问题表示子问题
- 得到平凡解
- 给出基于子问题表达式的算法
- 时间分析

- Computing binomial coefficients
- multistage graph problem
- Longest path in DAG
- Edit distance
- Knapsack
- Chain matrix multiplication
- all-pairs shortest paths
- Traveling salesman