4.6 The Minimum Edit Distance

- 问题定义
- 问题求解
  - 优化解的结构分析
  - 建立优化解代价的递归方程
  - 递归地划分子问题
  - 自底向上计算优化解的代价
    记录优化解的构造信息
  - 构造优化解

问题的定义

- 最小编辑距离

  输入：两个字符串$x[1..m]$和$y[1..n]$

  输出：将$x[1..m]$转换为$y[1..n]$所需要的最少操作数.

  操作：插入一个符号，或者
  删除一个符号，或者
  替换一个符号

  例如：x="snowy", y="sunny", "—"表示空字符

S	—	N	O	W		Y
S	U	N	N	—		Y

Cost: 3

—	S	N	O	W	—	Y
S	U	N	—	—	N	Y

Cost: 5

优化解结构分析

- 寻找优化解拆分成子问题解的方式

$ED[m,n]$：字符串$x[1{:}m]$和$y[1{:}n]$的编辑距离

$$ED[m,n]=\begin{cases}?\\ \dots\\ ?\end{cases}$$

根据$x[m]$的匹配位置分情况讨论

不能遗漏拆分方式

- 综合所有情况

X: …… X_m — … —
Y: … … … …… Y_n

如果$X[m]$匹配$Y[n]$之前的字符：

$$ED[m,n] = ED[m,n-1] + 1$$

X: … …… …… X_m
Y: … … … …… Y_n

如果$X[m]$匹配$Y[n]$：

$$ED[m,n] = ED[m-1,n-1] + diff(X[m],Y[n])$$

$$diff(x,y) = \begin{cases} 0 & if\ x = y \\ 1 & if\ x \neq y \end{cases}$$

如果$X[m]$匹配$Y[n]$之后的字符：

$$ED[m,n] = ED[m-1,n] + 1$$

上述三种情况涵盖所有无重叠操作的编辑序列

X: …… …… …… X_m
Y: …… Y_n — … —

优化子结构证明

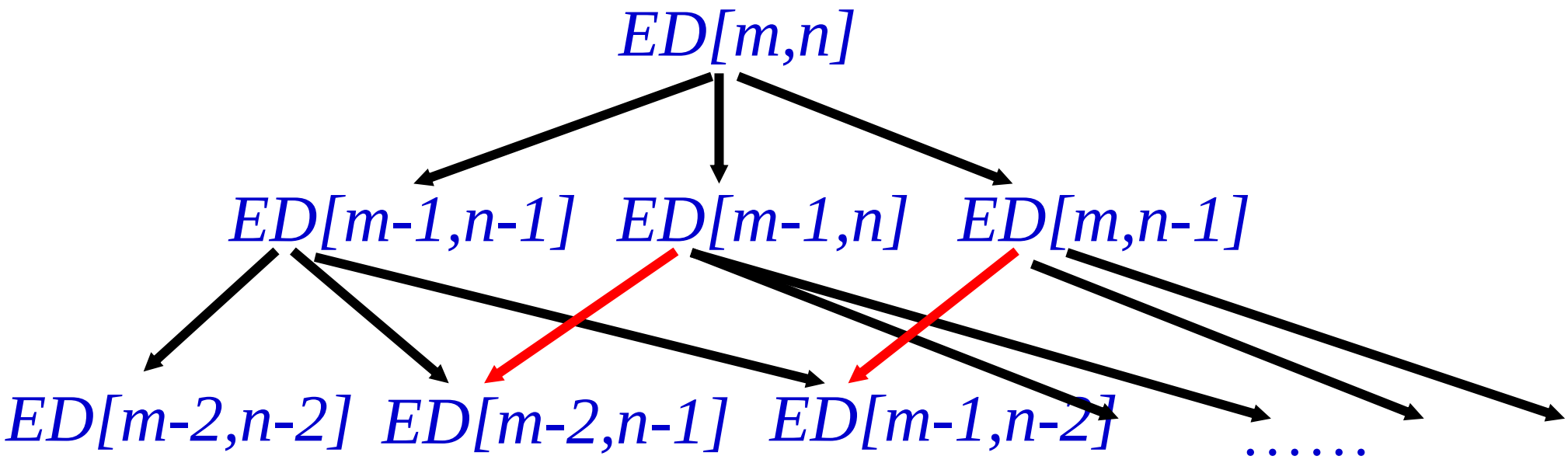
$$ED[m,n]=min\begin{cases}ED[m-1,n]+1\\ED[m-1,n-1]+diff(X[m],Y[n])\\ED[m,n-1]+1\end{cases}$$

证明：

情况一：$ED[m,n]=ED[m-1,n]+1$并取得优化解，则$ED[m-1,n]$必为$x[1:m-1]$和$y[1:n]$的最小编辑距离。否则，将存在一组编辑操作将$x[1:m-1]$在k步转换为$y[1:n]$，且$k<ED[m-1,n]$。如此，找到$k+1$次编辑操作将x转换为y，$k+1<ED[m,n]$，与$ED[m,n]$为优化解矛盾！

情况二、情况三的证明过程与情况一类似

子问题重叠性



最小编辑距离问题具有子问题重叠性

最小编辑距离的递归方程

- 计算$X[1:i]$和$Y[1:j]$的最小编辑距离$ED[i,j]$

$$ED[i,0] = i \qquad 1 \le i \le m$$

$$ED[0,j] = j \qquad 1 \le j \le n$$

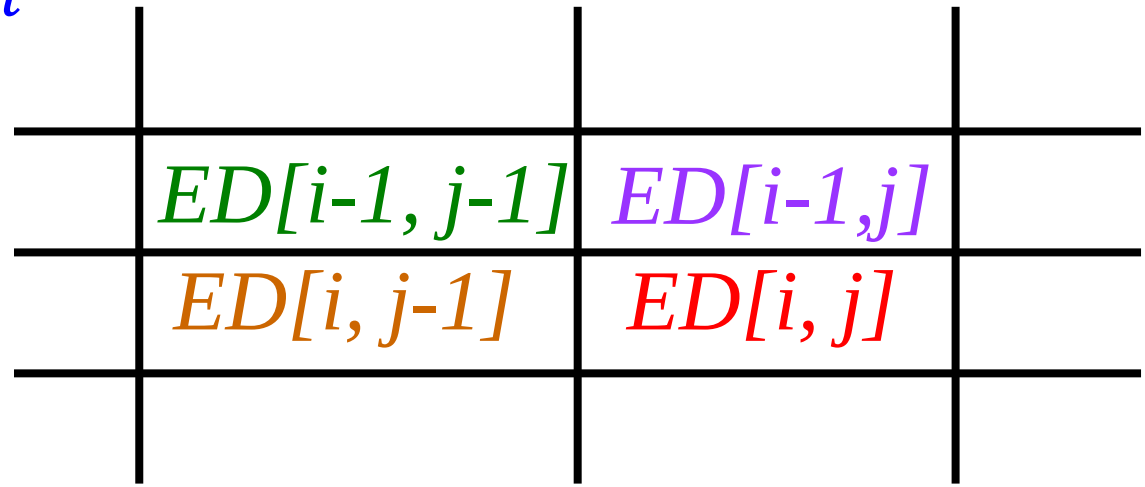
$$ED[i,j] = min\begin{cases} ED[i-1,j]+1 \\ ED[i-1,j-1]+diff(X[i],Y[j]) \\ ED[i,j-1]+1 \end{cases}$$

递归划分与自底向上求解

$$ED[i,j] = min\begin{cases} ED[i-1,j]+1 \\ ED[i-1,j-1]+diff(X[i],Y[j]) \\ ED[i,j-1]+1 \end{cases}$$

$$ED[i,0] = i \qquad 1 \le i \le m$$

$$ED[0,j] = j \qquad 1 \le j \le n$$



$$ED[i,j] = min\begin{cases} ED[i-1,j]+1 \\ ED[i-1,j-1]+diff(X[i],Y[j]) \\ ED[i,j-1]+1 \end{cases}$$

$i=0$	x_i \ y_j	R	E	N	A	T	O
	R						
	O						
	N						
	A						
	L						
	D						
	O						

$j=0$

记录优化解信息

$$ED[i,j]=min\begin{cases}ED[i-1,j]+1\\ED[i-1,j-1]+diff(X[i],Y[j])\\ED[i,j-1]+1\end{cases}$$

		y_j	R	E	N	A	T	O
$i=0$	x_i	0	1	2	3	4	5	6
R		1	↖ 0	←1	←2	←3	←4	← 5
O		2	↑ 1	↖1	↖2	↖3	↖4	↖ 4
N		3	↑ 2	↖2	↖1	←2	←3	←4
A		4	↑ 3	↖3	↑2	↖1	←2	←3
L		5	↑ 4	↖4	↑3	↑2	↖2	↖3
D		6	↑ 5	↖5	↑4	↑3	↖3	↖3
O		7	↑ 6	↖6	↑5	↑4	↖4	↖3

$j=0$

记录优化解信息

$$ED[i,j] = min\begin{cases} ED[i-1,j]+1 \\ ED[i-1,j-1]+diff(X[i],Y[j]) \\ ED[i,j-1]+1 \end{cases}$$

	y_j	R	E	N	A	T	O
$i=0$	x_i	0	1	2	3	4	5	6
R	1	↖ 0	←1	←2	←3	←4	← 5
O	2	↑ 1	↖1	↖2	↖3	↖4	↖ 4
N	3	↑ 2	↖2	↖1	←2	←3	←4
A	4	↑ 3	↖3	↑2	↖1	←2	←3
L	5	↑ 4	↖4	↑3	↑2	↖2	↖3
D	6	↑ 5	↖5	↑4	↑3	↖3	↖3
O	7	↑ 6	↖6	↑5	↑4	↖4	↖3

$j=0$

```
MinimumED(X, Y)
M←length(X); n←length(Y);
For i←0 To m Do
        E[i,0]←i;
For j←1 To n Do
        E [0,j]←j;
For i←1 To m Do
    For j←1 To n Do
        If x_i = y_j
            Then E[i,j]←E[i-1,j-1];
        Else
            E[i,j]←E[i-1,j-1]+1;
        B[i,j]←"↖";
        If E[i,j]>E[i-1,j]+1
            Then E[i,j]=E[i-1,j]+1;
            B[i,j]←"↑";
        If E[i,j]>E[i,j-1]+1
            Then E[i,j]=E[i,j-1]+1;
            B[i,j]←"←";
Return E and B.
```

算法和算法复杂性

- 时间复杂性
  - (i, j)两层层循环，每层循环至多m和n步
  - 时间复杂性为$O(mn)$
- 空间复杂性
  - 一个$(m+1)\times(n+1)$数组，一个$m\times n$数组
  - $O(mn)$
  - B可以省去

构造优化编辑序列

$$ED[i,j]=min\begin{cases}ED[i-1,j]+1\\ED[i-1,j-1]+diff(X[i],Y[j])\\ED[i,j-1]+1\end{cases}$$

		y_j	R	E	N	A	T	O
i=0	x_i	0	1	2	3	4	5	6
R		1	↖0	←1	←2	←3	←4	←5
O		2	↑1	↖1	←2	←3	←4	↖4
N		3	↑2	↑2	↖1	←2	←3	←4
A		4	↑3	↑3	↑2	↖1	←2	←3
L		5	↑4	↑4	↑3	↑2	↖2	↖3
D		6	↑5	↑5	↑4	↑3	↖3	↖3
O		7	↑6	↑6	↑5	↑4	↖4	↖4

j=0

边界条件$E[i,0]$：删除$x[1:i]$

边界条件$E[0,j]$：在$x[1]$前插入$y[1:j]$

↖ 将$x[1:i-1]$修改为$y[1:j-1]$后，将$x[i]$按需修改为$y[j]$

← 将$x[1:i]$修改为$y[1:j-1]$后，在末尾插入$y[j]$

↑ 将$x[1:i-1]$修改为$y[1:j]$后，删除末尾符号（原$x[i]$）

问题的定义

- 最小编辑距离判别问题

  输入：两个字符串$x[1..m]$和$y[1..n]$，整数t

  输出：$True$，如果x和y的最小编辑距离不大于t

  $False$，如果x和y的最小编辑距离大于t

输入：RONALDO, RENATO, 4, 输出：$True$

输入：RONALDO, RENATO, 2, 输出：$False$

问题的定义

- 最小编辑距离判别问题

  Step 1. 计算输入字符串的最小编辑距离

  Step 2. 与阈值比较

在E中计算大量不必要的元素！

- 改进算法的两点启发

$$ED[i,j] = min\begin{cases} ED[i-1,j]+1 \\ ED[i-1,j-1]+diff(X[i],Y[j]) \\ ED[i,j-1]+1 \end{cases}$$

$$ED[i,0] = i \qquad 1 \le i \le m$$

$$ED[0,j] = j \qquad 1 \le j \le n$$

字符串x和y的最小编辑距离不小于二者长度之差的绝对值

x="RONALDO", y="RENATO",

编辑距离阈值$t = 2$

$i=0$	x_i		R	E	N	A	T	O
	R							
	O							
	N							
	A							
	L							
	D							
	O							

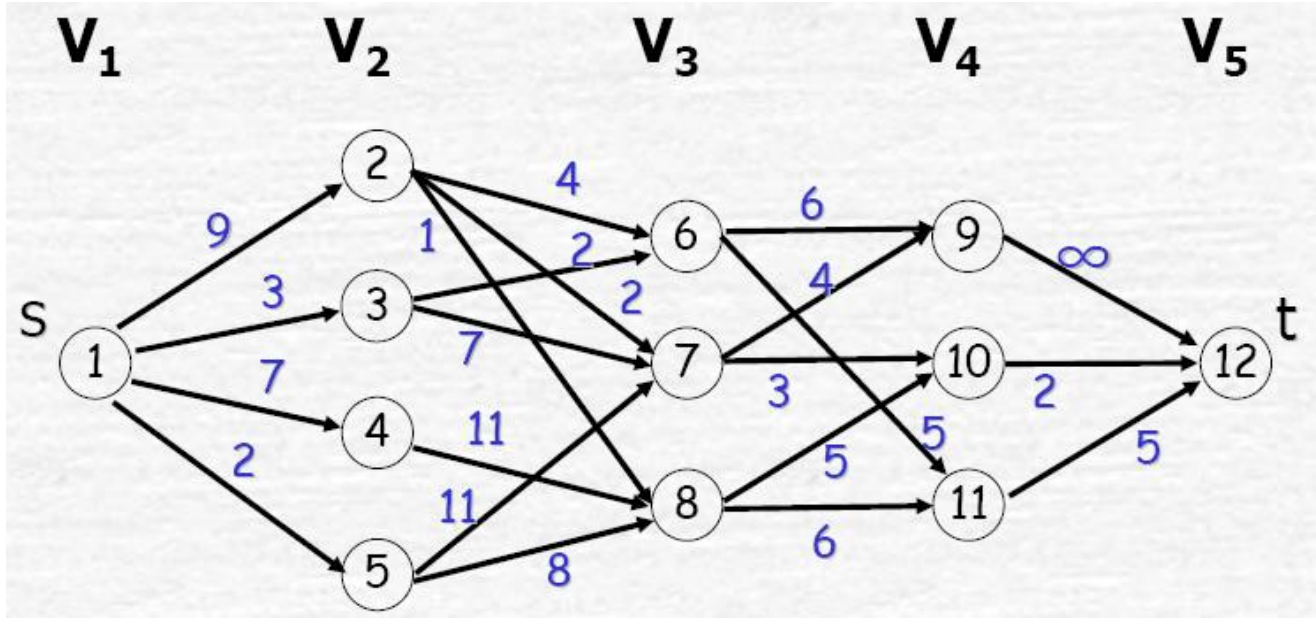
算法的复杂性

x="RONALDO",
y="RENATO",
编辑距离阈值$t = 2$

i=0	x_i	R	E	N	A	T	O
		0	1	2				
R		1	0	1	2			
O		2	1	1	2	3		
N			2	2	1	2	3	
A				3	2	1	2	3
L					3	2	2	3
D						3	3	3
O								

- 时间复杂性
  - 每行、列最多计算$2t + 1$
  - 时间复杂性为$O(min\{m,n\}t)$
- 空间复杂性
  - $O(min\{m,n\}t)$
  - 可优化为$O(t)$

多段图规划



求从s到t的最短路径.

多段图规划

- 优化解的结构分析

设$s,\ldots,v_{ij},\ldots,v_{ik},\ldots,t$是一条由$s$到$t$的最短路径，则$v_{ij},\ldots,v_{ik},\ldots,t$也是由$v_{ij}$到$t$的最短路径

子集求和

- 问题定义

  输入：集合$S=\{n$个正整数$\}$，正整数P

  输出：True，若存在子集合S'，使得$P=S'$中所有数之和

  False，否则

子集求和

- 优化解结构分析

$M(i, j)$ ：$=True$，当且仅当在S的前i个数中，存在一个子集合，使其数据之和为j.

$M(n, P)$即为原始问题

$$M(i, j) = M(i-1, j-S[i]) \vee M(i-1, j)$$

$$M(i, 0) = True$$
$$M(0, j) = False \qquad for\ j>0$$

思考题

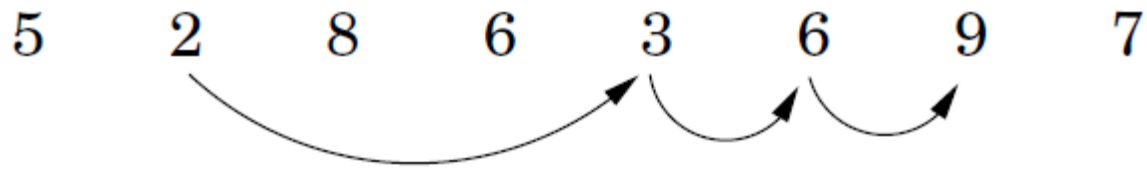
- 最长增长子序列问题

  输入：由n个数组成的一个序列S：$a_1,a_2,\ldots,a_n$

  输出：子序列$S'=b_1,b_2,\ldots,b_k$，满足：

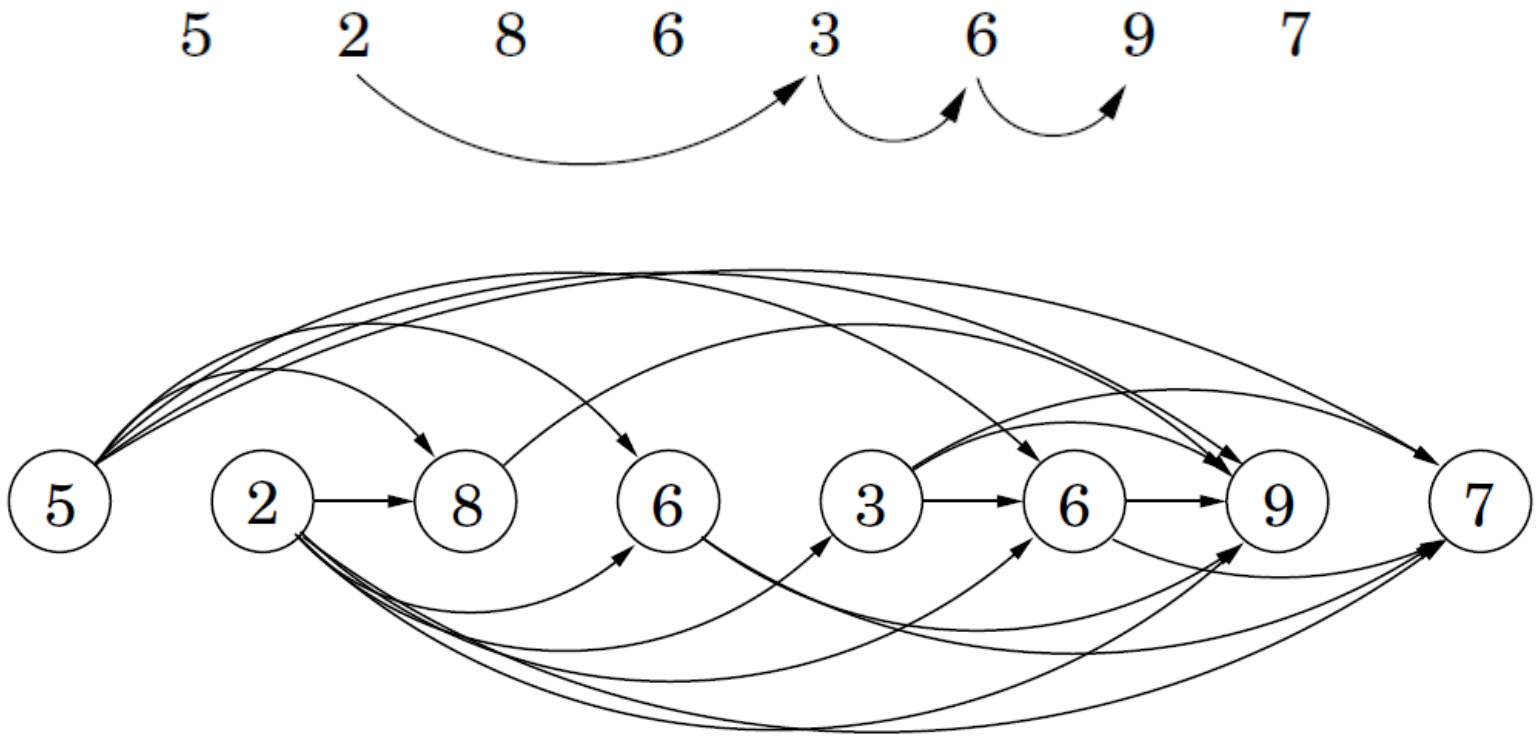
  (1) $b_1 \le b_2 \le \ldots \le b_k$，

  (2) $|S'|$最大



思考题

- 最长增长子序列问题



总结

- 原始问题可以划分成一系列子问题，子问题之间不是相互独立的
- 不同子问题的数目常常只有多项式量级
- 优化子结构

总结

- 优化解的结构分析
- 建立优化解代价的递归方程
- 递归地划分子问题
- 自底向上计算优化解的代价
  记录优化解的构造信息
- 构造优化解