

# A new distance

---

凌嘉伟

151220061

# 上一节课最后留下一个问题

---

- 你还有哪些创意，来定义二进制位串之间的距离？

# 定义一个新的距离

- 定义距离：两个二进制串之间由一个转成另一个所需的最少编辑操作次数
- 编辑操作包括：删除某一位、在某一位前后添加0或1、修改位串中的某一位
- 例如：1010和1101，它们的汉明距离为3，而这样表示的距离为2  
(1010-101-1101)

- 
- 其实这样表示的距离已经存在，最初由俄罗斯人莱文斯坦提出，被称作莱文斯坦距离，是编辑距离问题的一种，我们可以将二进制位串之间的距离扩展为求两个字符串之间的编辑距离。
  - 例如kitten和sitting: kitten-sitten-sittin-sitting

# 如何求这样的距离

- 可以用动态规划的方法求：
- $f(i,j)$ 表示长度为*i*的字符串到长度为*j*的字符串的编辑距离

$$f(i,j) = \begin{cases} 0 & i = 0 \text{ and } j = 0, \\ j & i = 0 \text{ and } j > 0, \\ i & i > 0 \text{ and } j = 0, \\ \min\{f(i-1,j) + 1, f(i,j-1) + 1, f(i-1,j-1) + a(i,j)\} & i \geq 1 \text{ and } j \geq 1. \end{cases}$$

$$a(i,j) = \begin{cases} 1 & str1(i) \neq str2(j), \\ 0 & str1(i) = str2(j). \end{cases}$$

- `int distance(string str1,string str2)`
- `{`
  - `for (i = 0;i < str1.size();i++) dist[i][0] = i;`
  - `for (i = 0;i < str2.size();i++) dist[0][i] = i;`
  - `for (i = 0;i < str1.size();i++)`
    - `for (j = 0;j < str2.size();j++)`
      - `{`
        - `if (str1[i] != str2[j]) cost = 1; else cost = 0;`
        - `dist[i][j] = min(...);`
      - `}`
    - `return dist[str1.size()][str2.size()];`
    - `}`



# 汉明距离扩展——李距离

- 在编码理论中，李距离是评价两个长度为 $n$ 的 $q$ 进制字符串 $x_1x_2\cdots x_n$ 和 $y_1y_2\cdots y_n$ 之间的距离， $q > 1$ ，距离为

$$\sum_{i=1}^n \min\{|x_i - y_i|, q - |x_i - y_i|\}$$

- 例如： $q = 6$ ，3340和2543之间的距离为 $1 + 2 + 0 + 3 = 6$
- 当 $q = 2$ 或 $3$ 时等价于汉明距离

- 
- Q&A



---

- 谢谢大家！