# 从立体几何问题看降低编程复杂度

人大附中 高亦陶

# 引言：一类立体几何问题

- O(1) 的空间复杂度
- O(1) 的时间复杂度
- 并非公认的简单题

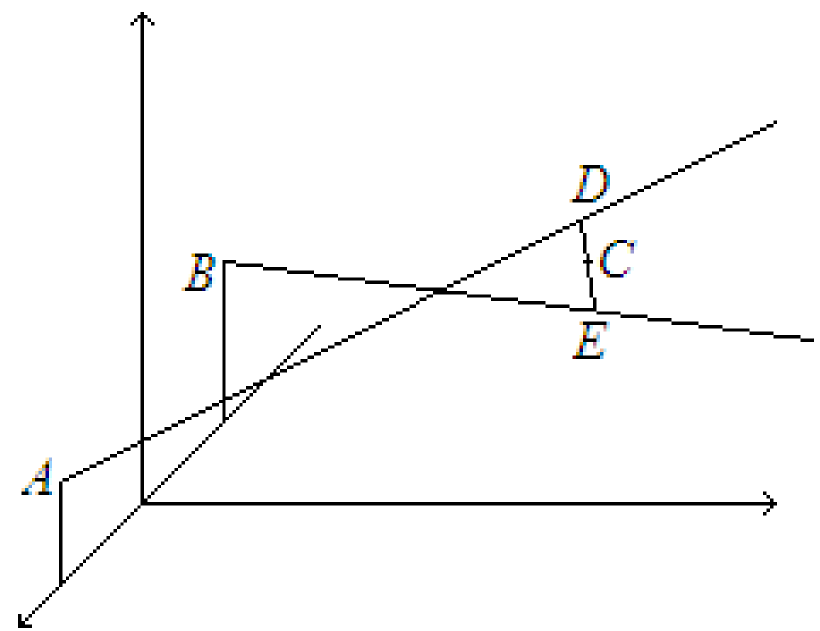
巨大的编程复杂度

# 1 运用合适的思维方式

- 使用方程是一种进步
  方程是一种抽象的、通用的解题方法
  但是方程有时会忽略一部分已知信息
- 通过具体地思考、充分利用已知信息可以从本质入手，降低编程复杂度

# 例 1 Model Rocket Height

- 给出两条直线的起点和方向，求它们公垂线中点的高度。
- 直线方向用仰角和方向角表示。

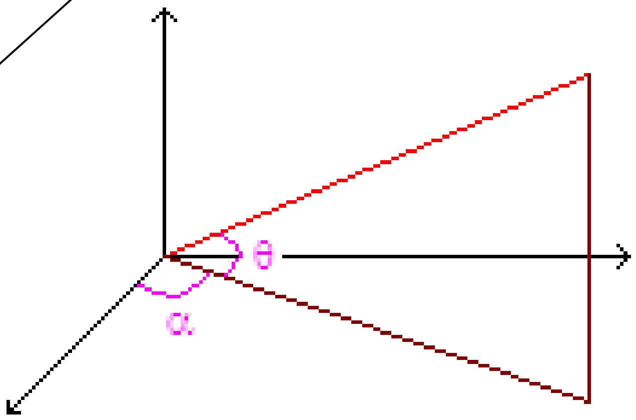


# 数据的初步处理和思路

公垂线 ← 叉积 ← 方向向量

$$\mathbf{n} = \mathbf{a} \times \mathbf{b}$$

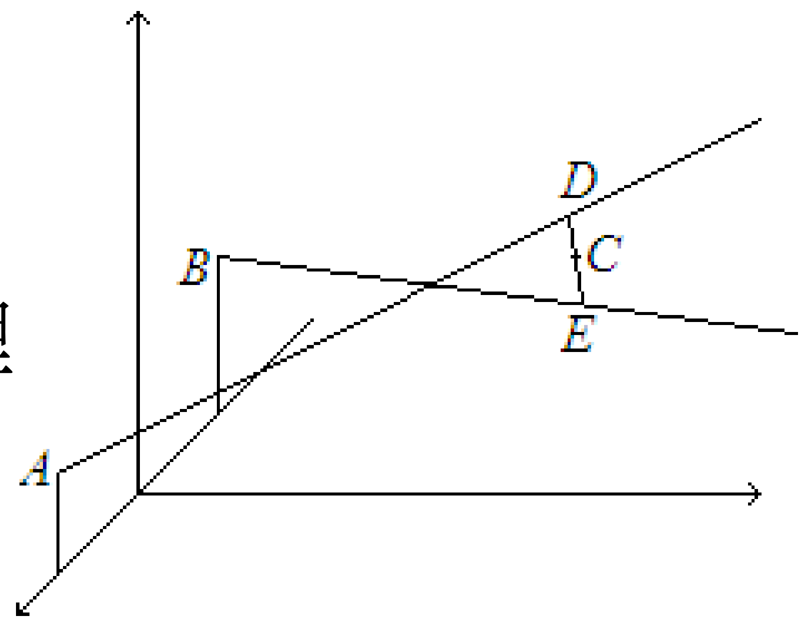
$$\left(1, \tan\alpha, \sec\alpha \tan\theta\right)$$



# 尝试解题

$$\overrightarrow{AB} = \overrightarrow{AD} + \overrightarrow{DE} - \overrightarrow{BE}$$

$$= \lambda_1 \cdot \mathbf{a} + \lambda_3 \cdot \mathbf{n} - \lambda_2 \cdot \mathbf{b}$$




根据空间向量基本定理有唯一解

可以化成

三元一次方程组

消元? 行列式?

浮点误差?

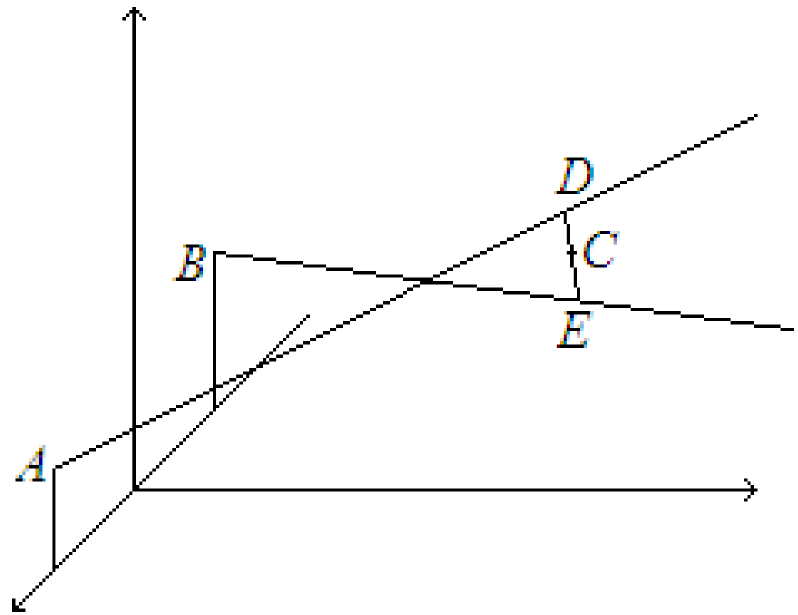
**麻烦**

# 进一步思考

$$\overrightarrow{AB}=\overrightarrow{AD}+\overrightarrow{DE}-\overrightarrow{BE}$$

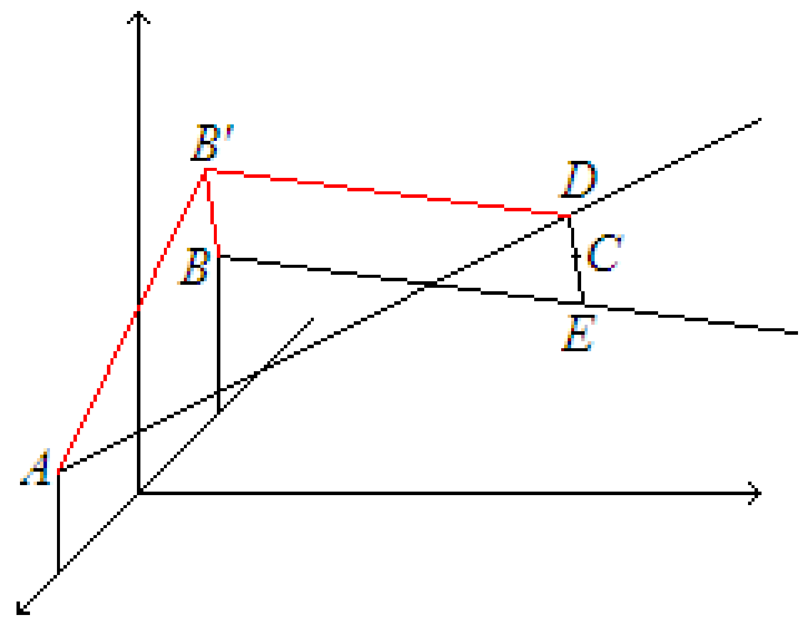
$$\overrightarrow{AB}\times\mathbf{n}=\overrightarrow{DE}\times\mathbf{n}=\left|\overrightarrow{DE}\right|\left|\mathbf{n}\right|$$

$$\overrightarrow{DE}=\frac{\left|\overrightarrow{DE}\right|}{\left|\mathbf{n}\right|}\times\mathbf{n}=\frac{\overrightarrow{AB}\times\mathbf{n}}{\mathbf{n}^{2}}\times\mathbf{n}$$



# 另外两个未知量

$$\overrightarrow{AB'} = \overrightarrow{AB} - \overrightarrow{DE} = \overrightarrow{AD} - \overrightarrow{BE}$$




三角形内已知一边和内角，求另一边长

# 最后一步

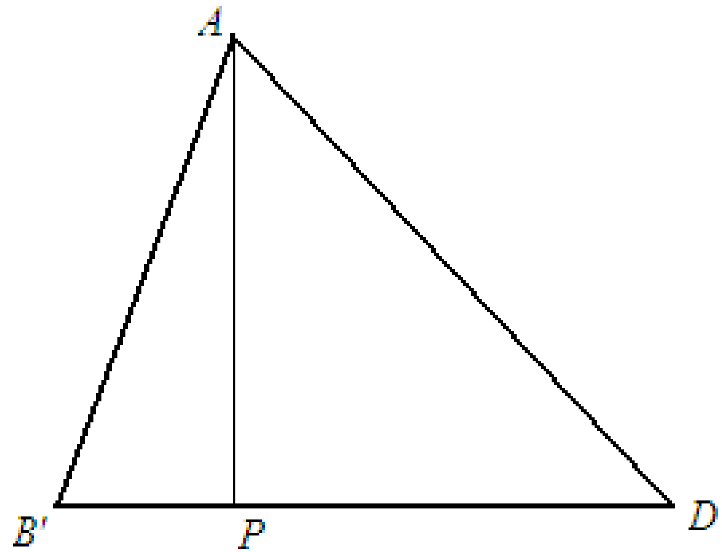
$$\overrightarrow{AB'}\cdot\mathbf{b}=\overrightarrow{PB'}\cdot\mathbf{b}=\left|\overrightarrow{PB'}\right|\left|\mathbf{b}\right|$$

$$\overrightarrow{PB'}=\frac{\left|\overrightarrow{PB'}\right|}{\left|\mathbf{b}\right|}\cdot\mathbf{b}=\frac{\overrightarrow{AB'}\cdot\mathbf{b}}{\mathbf{b}^{2}}\mathbf{b}$$

$$\left|\overrightarrow{AD}\right|\cos\angle DAP=\left|\overrightarrow{AP}\right|$$

$$\overrightarrow{AB}=\overrightarrow{AB'}+\overrightarrow{PB'}$$

$$\overrightarrow{AD}=\frac{\left|\overrightarrow{AD}\right|}{\left|\mathbf{a}\right|}\cdot\mathbf{a}=\frac{\left|\overrightarrow{AP}\right|}{\dfrac{\mathbf{a}\cdot\overrightarrow{AP}}{\left|\mathbf{a}\right|\cdot\left|\overrightarrow{AP}\right|}\cdot\left|\mathbf{a}\right|}\mathbf{a}=\frac{\overrightarrow{AP}^{2}}{\mathbf{a}\cdot\overrightarrow{AP}}\mathbf{a}$$



# 小结

$$\overrightarrow{DE}=\frac{\overrightarrow{AB}\times\mathbf{n}}{\mathbf{n}^{2}}\times\mathbf{n}$$

$$\overrightarrow{AB'}=\overrightarrow{AB}-\overrightarrow{DE}$$

$$\overrightarrow{PB'}=\frac{\overrightarrow{AB'}\times\mathbf{b}}{\mathbf{b}^{2}}\times\mathbf{b}$$

$$\overrightarrow{AP}=\overrightarrow{AB'}-\overrightarrow{PB'}$$

$$\overrightarrow{AD}=\frac{\overrightarrow{AP}^{2}}{\mathbf{a}\times\overrightarrow{AP}}\times\mathbf{a}$$

$$\overrightarrow{OC}=\overrightarrow{OA}+\overrightarrow{AD}+\frac{1}{2}\overrightarrow{DE}$$

将盲目的方程组求解
改为一系列向量运算

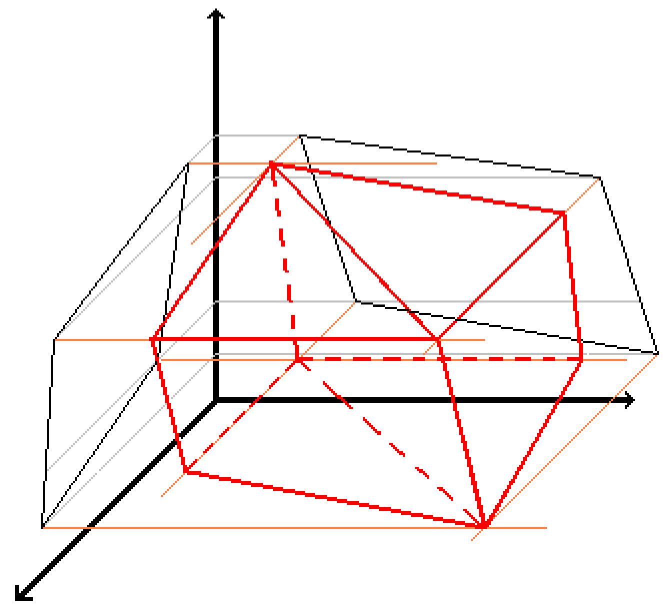
降低了编程复杂度

# 2 注意分类讨论

- 大量的分类 + 复杂的判断 = 难以承受的编程复杂度
- 合理地把不同的情况合并起来可以大大改善这种状况

# 例 2 Crossing Prisms

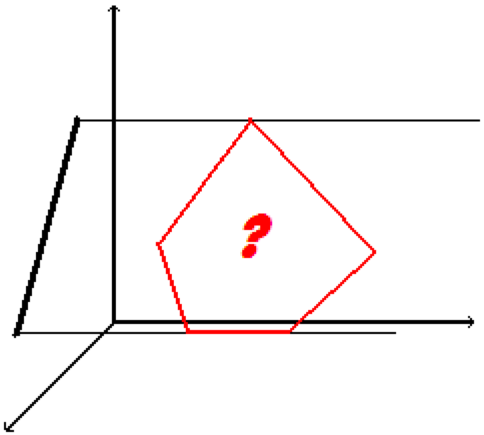
- 平面内有一个简单多边形
- 现在将这个多边形分别放置在 xz 面和 yz 面
- 多边形边数 ≤ 4，每个顶点的 x 坐标都是整数，坐标取值范围为 [0,10]
- 求这两个棱柱的交……按逆时针排序。



# 关注表面

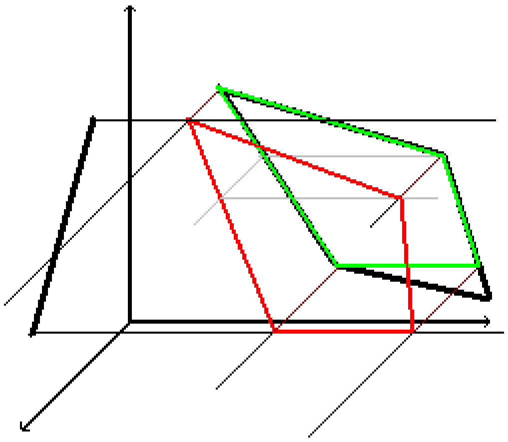
观察某个柱的一个侧面

它的一部分是交的表面



# 多数情况

如果侧面与底面不平行



交的表面一定是
用侧面截柱得到的截面

面积 ×cos 二面角 = 射影面积

二面角很容易求

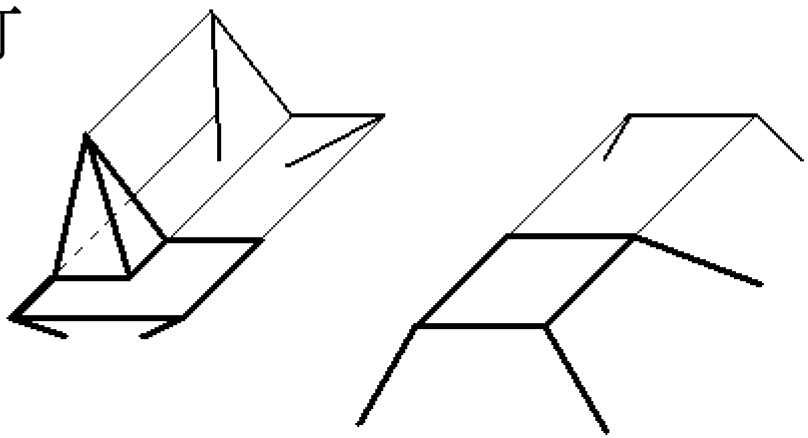
射影面积＝柱底图形与 y1 ≤ y ≤ y2 的交的面积

# 重要情况

如果侧面与底面平行

边数≤ 4

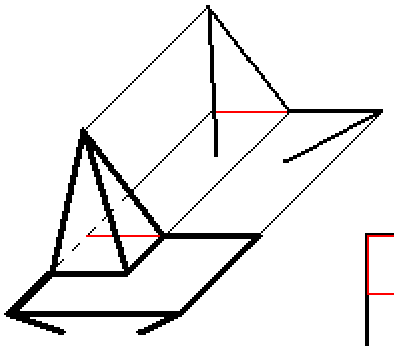
可以证明
只有图中两种情况

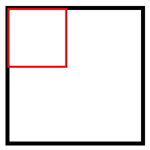


# 形状特殊的面

柱底图形
在特定高度上的宽

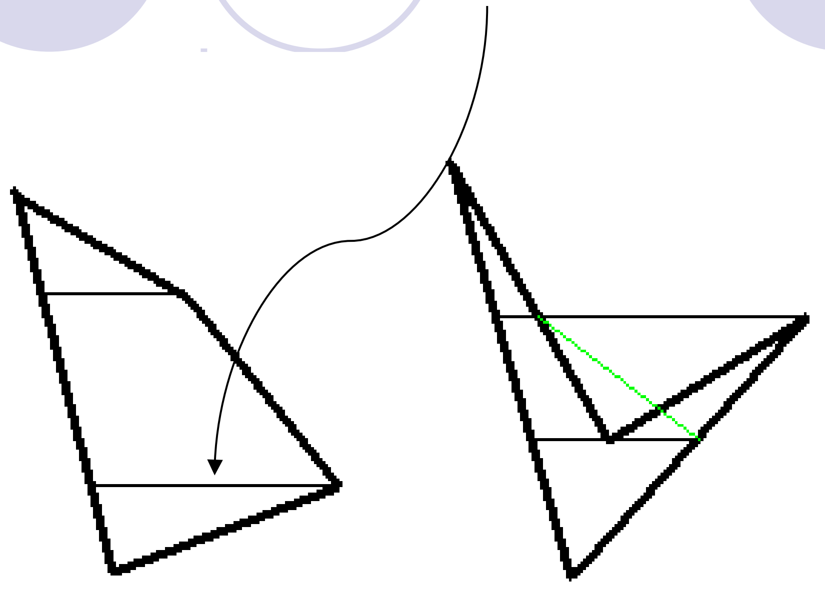
面积 = 宽 $^2$-( 宽 - 边长 )$^2$





对正方形也适用！

# 继续利用这个宽



也可以用来计算射影面积！

射影面积 = sum((宽 $_1$+ 宽 $_2$)* 高 /2)

需要对高度排序

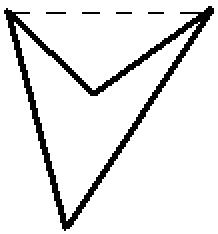
# 算法框架

- 对高度排序
- 计算每点高度处的宽
- 枚举每一条边
  - 判断平行与否
  - 宽 $^2$-( 宽 - 边长 )$^2$
  - 或者　2* 射影面积 /cos 二面角

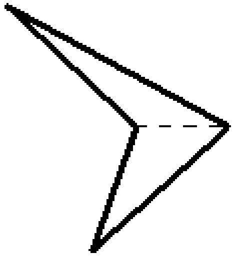
# 计算宽 处理特殊情况

求所有边与 $y=y_k$ 的交点

最大值 - 最小值？



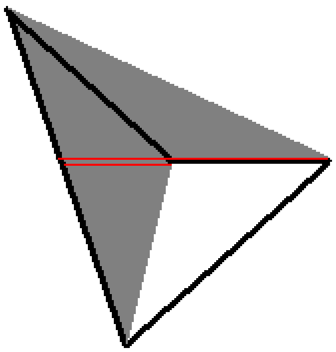
完全不考虑不规范交点？



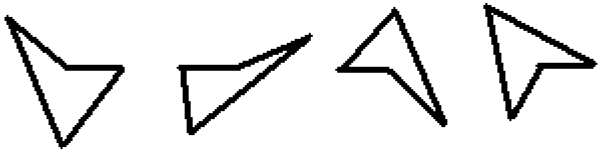
利用逆时针顺序关系确定交点方向

# 面积≠(宽$_1$+宽$_2$)*高/2？

特判会打破先前的算法框架



在局部改变宽的定义，利用点的逆时针序忽略一些边，使两个宽不同

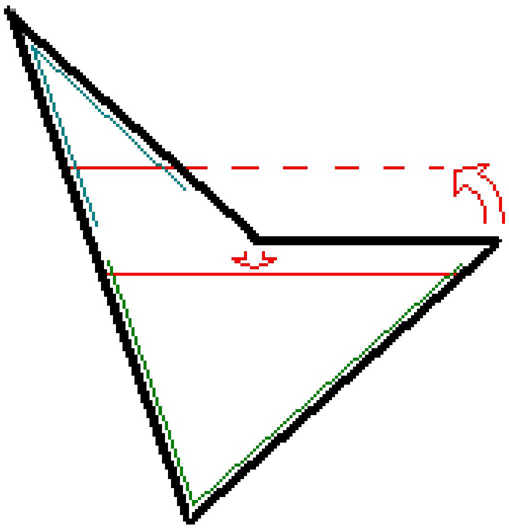


修改两个点的高度顺序最终使得面积可以照常计算

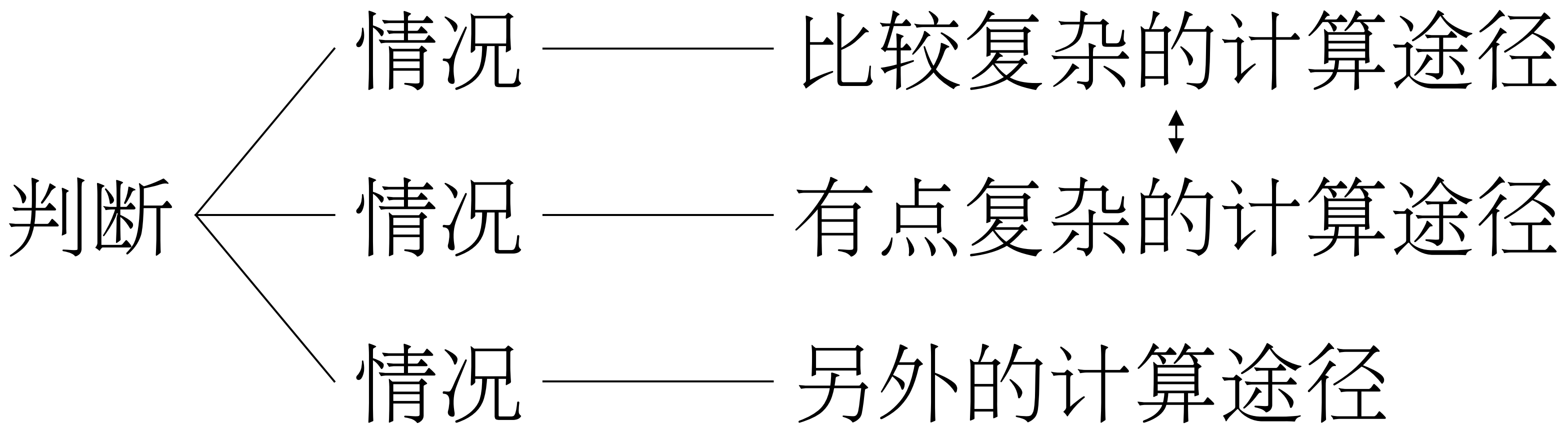
# 一种具体的处理办法

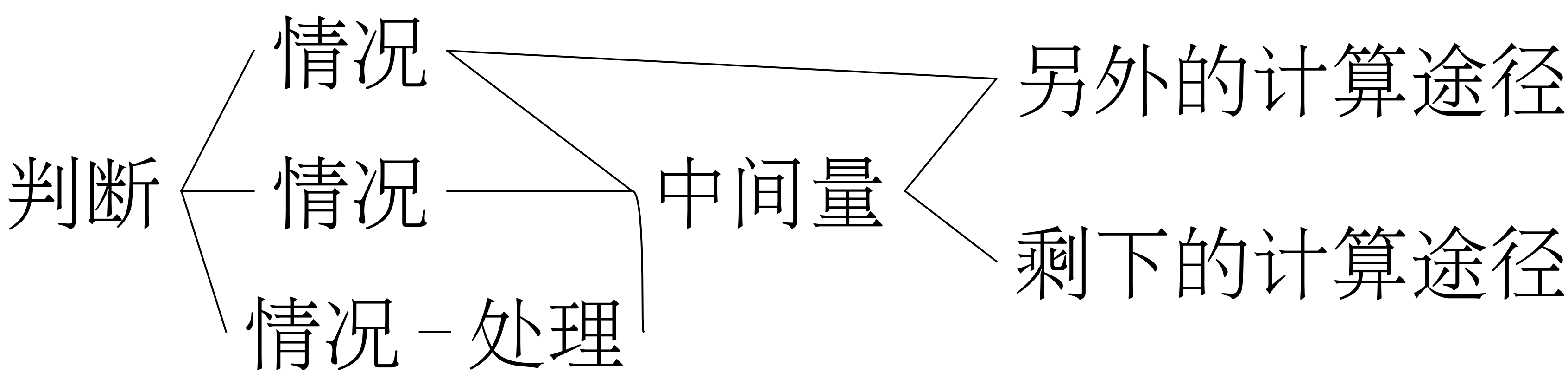
忽略和每个点相邻的边，让凹角顶点对应的宽较大

同时确保四个点的高按逆时针顺序呈 1,2+,2-,3 或 3,2-,2+,1 的形式



# 小结：合并的效果


判断
情况
比较复杂的计算途径
情况
有点复杂的计算途径
情况
另外的计算途径


判断
情况
情况
情况
处理
中间量
另外的计算途径
剩下的计算途径

# 总结和启示

- 算法是多样化的，选择时要注重适用性
- 在遇到新问题时，首先想一想能不能在现有框架内解决，而不是随意添加新的内容
- 对算法同样可以从类似内容中合并相同点从而达到精简的效果

谢谢