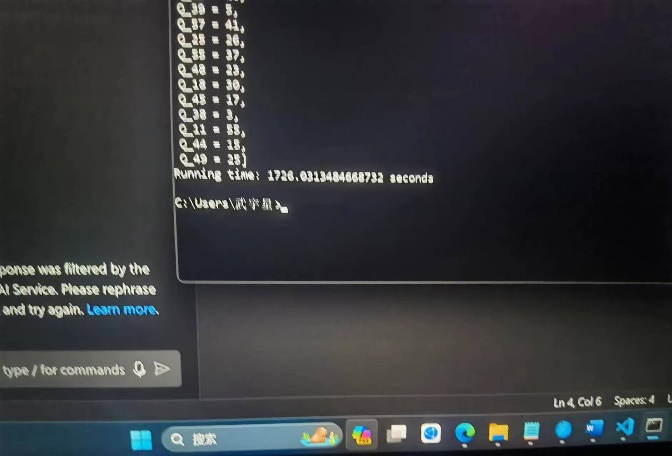
N\_Queens实验记录如下

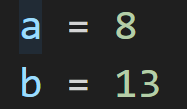
|  |  |  |
| --- | --- | --- |
| N | SMT\_RunningTime(s) | SAT\_RunningTime(s) |
| 8 | 0.04614686965942383 | 0.1271669864654541 |
| 10 | 0.05881667137145996 | 0.22541356086730957 |
| 20 | 0.5753731727600098 | 1.8096275329589844 |
| 30 | 4.004112005233765 | 6.115971565246582 |
| 40 | 17.451601266860962 | 15.567112684249878 |
| 50 | 63.26491379737854 | 28.994419813156128 |
| 60 | 1726 | 57.56117129325867 |



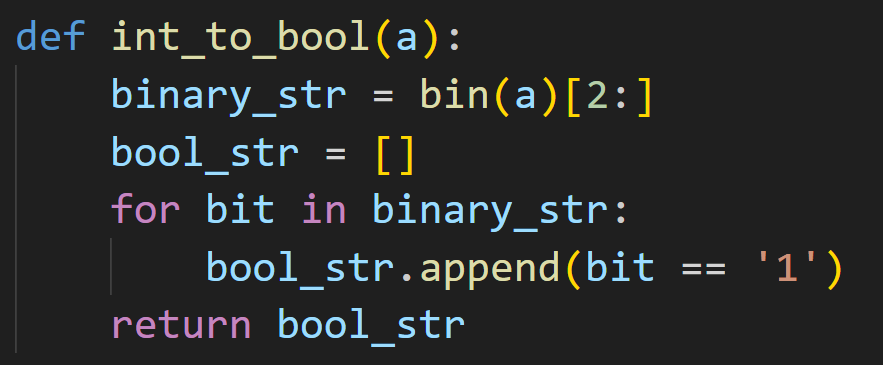
当N小时SMT的效率要高于SAT，但当N>=40时SAT效率更高，我认为这是因为在添加约束时SAT的时间复杂度是O(n^3),而SMT为O(n)，这造成了SAT在N小时效率低的情况，而若仅仅从solve前记录时间会发现SAT更快

D = a – b

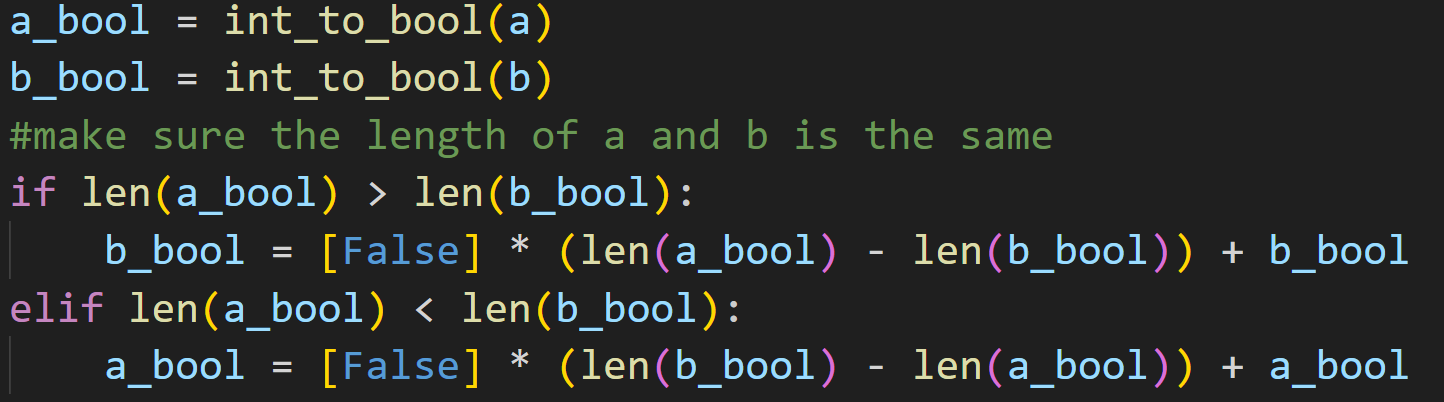
在程序中预先定义好ab的值



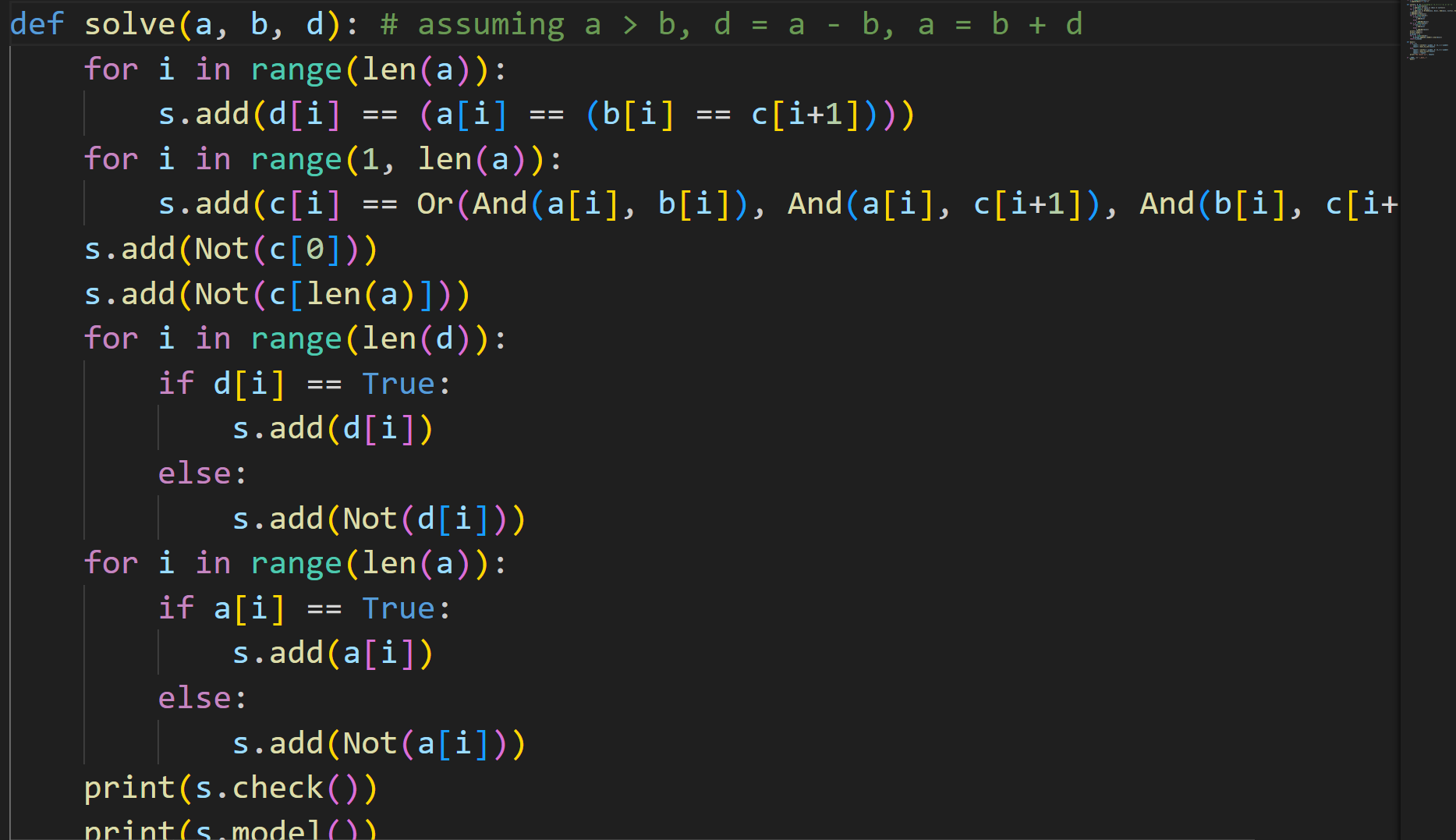
首先需要对整数进行预处理转化为bool即pure SAT



同时也需确保ab长度一致方便计算

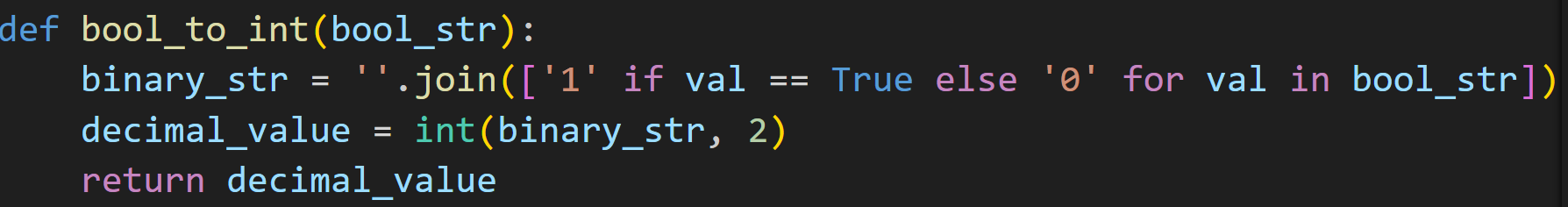


在完成预处理后即可根据PPT上加法部分完成SAT约束

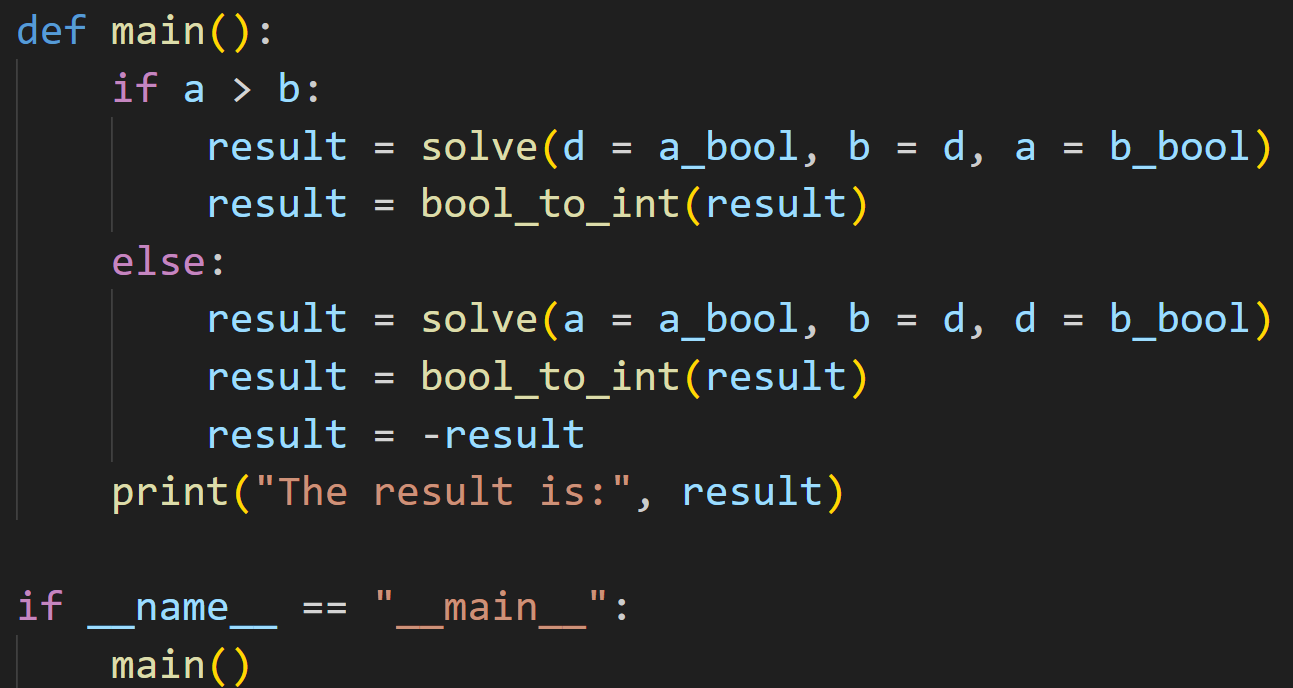


需要注意的是与PPT不同，abc的index均从0开始，因此需要右对齐，故约束中为c[i+1]而非c[i]，同时为实现减法应给定a的约束和d的约束而非a和b的约束

同时也需要函数将bool转化为整数

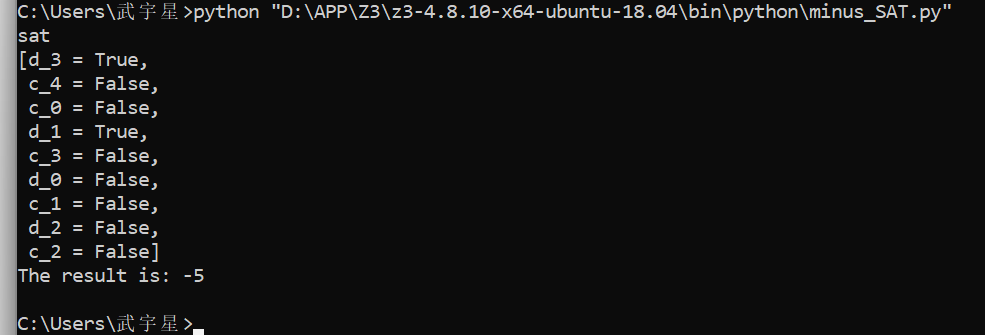


在main部分只需判断ab大小关系，若a小则计算b-a最后取反即可



运行结果如下：

a = 8, b = 13



A = 15 b = 13

