**مائده نادهی 9712762238**

**مدلسازی**

هر خانه ی 0 موجود در آرایه ی دو بعدی دریافتی از کاربر یک متغیر درنظر گرفته میشود. (خانه های سفید رنگ)

خانه های پازل میتوانند از نوع Cell (خانه های سیاه) یا Constraint (خانه های سبز) یا Variable (خانه های سفید) باشد کلاس problem جدول پازل را نگهداری میکند و برای صرفه جویی در حافظه متد back\_track فقط از یک آبجکت problem که به صورت global تعریف شده استفاده میکند (ینی تغییرات اعمال شده روی برد بازی را هر باز پس از فراخوانی بازگشتی متد back\_track خدف میکند (undo))

**نکات مهم کلاس Variable**

هر متغیر برای دسترسی سریع تر Constraint افقی و عمودی مرتبط با خود را در خود نگهداری میکند

استک domain\_history برای حفظ تغییرات در domain متغیر که در FC بوجود میاید است به این صورت که در هربار که FC فراخوانی میشود دامنه ی قبلی در این استک پوش میشود و در FC\_undo اخرین دامنه ی اضافه شده به استک پاپ میشود و به عنوان دامنه ی جدید ست میشود

**نکات مهم کلاس Constraint**

برای افرایش سرعت یک HashSet به نام valueOfRelatedVariables در این کلاس وجود دارد که value های متغیر های مرتبط با این constraint را درخود نگه میدارد (منظور از متغیر های مرتبط متغیر های پایین یک constraint عمودی و متغیر های سمت راست یک constraint افقی است)

متغیر lowerBound کمترین value ایی که متغیر های مرتبط با این constraint میتوانند داشته باشند را نگهداری میکند و به طور مشابه upperBound بیشترین value

(این دو متغیر بالا در node consistency مورد استفاده قرار میگیرند)

مقدار lowerBound به این صورت محاسبه می شود که یک متغیر را خالی درنظر گرفته و مابقی متغیر هارا 9و8و7و...

یعنی :

sum = ((20 - numberOfUnassignedVar) \* (numberOfUnassignedVar - 1)) / 2; *// sum of : 9, 8, 7, ..., (9-(numberOfRelatedVar-1)+1)*lowerbound = Math.*max*(currentValue - sum, 1);

upperBound هم به طور مشابه

**Node consistency**

باتوجه به upperBound و lowerBound در constraint های افقی و عمودی اش مشخص میشود

**MRV**

متغیر با کم ترین value محاز را برمیگداند (کمترین domain.size)

**LCV**

Domain هر متغیر را به ترتیب صعودی در constraint ای که با بقیه متغیر ها دارد مرتب میکند

**AC\_3**

مطابق شبه کد موجود در اسلاید ها

**FC**

از دامنه ی متغیر های هم سطر و هم ستون مقدار value انتخاب شده را حذف میکند و با توجه به مقادیر جدید lowerBound و upperBound آن را محدود میکند

پس از فعالسازی arc\_consistency زمان اجرای کاهش پیدا کرد.

تمامی مراحل قبل مورد نیاز نیست اما عدم وجود آنها باعث کاهش سرعت الگوریتم می شود.