**سوال پنجم:** بر مبنای پوشش مسیرهای پرایم، موارد آزمون را طراحی کنید. سپس کد کلاس StringToIntTest را تکمیل کنید. در نهایت خطا برنامه داده شده را مشخص کنید.

با داشتن مسیرهای پوشش پرایم که در قسمت قبلی بدست آمد، می‌توانیم مسیرهای آزمون را طراحی کنیم. مسیرهای آزمون می‌بایست از یک نود آغازین شروع و به یک نود پایانی ختم شوند (در صورتی که مسیرهای پوشش پرایم لزوما این شرط را ندارند). لذا می بایست مجموعه مسیر های آزمونی ایجاد کنیم که هر مسیر آزمون می‌بایست یک یا چند مسیر پرایم را به عنوان زیر مسیر در خود داشته باشد. هر چه تعداد مسیرهای آزمون ایجاد شده (و به تبع آن تعداد موارد آزمون تولید شده) کمتر باشد (با شرط پوشش کمتر یا مساوی با پوشش ایجاد شده توسط مسیرهای پرایم)، کارایی آزمون بیشتر خواهد شد.

پس از تحلیل مسیرهای پرایم و تلاش برای تبدیل آن‌ها به مسیرهای آزمون، مسیرهای یکتا طبق جدول زیر

|  |  |  |
| --- | --- | --- |
| شماره | مسیر آزمون | شماره مسیرهای پرایم پوشش داده‌شده |
| ۱ | 1,2,3 | ۲۱ |
| ۲ | 1,2,4,5,4,6,8,9,11,8,12 | ۴، ۹، ۱۳، ۱۴، ۱۸، ۲۰ |
| ۳ | 1,2,4,5,4,6,7,8,12 | ۵، ۱۱، ۱۳، ۲۰ |
| ۴ | 1,2,4,5,4,6,8,12 | ۸، ۱۲، ۱۳، ۲۰ |
| ۵ | 1,2,4,5,4,6,7,8,9,11,8,9,11,8,9,10 | ۲، ۳، ۶، ۷، ۱۰، ۱۳، ۱۵، ۱۶، ۱۷، ۱۸، ۱۹، ۲۰ |

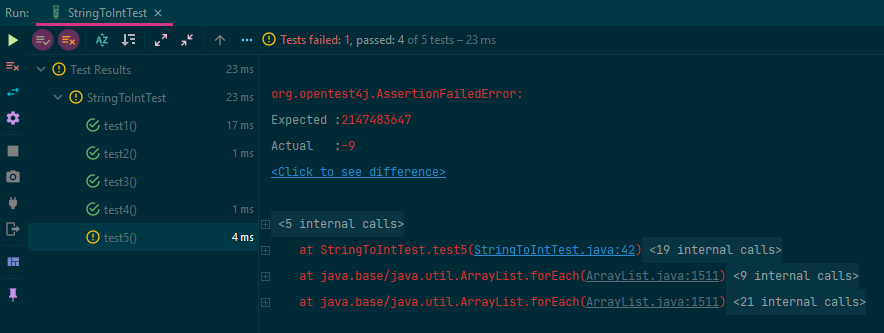
بدست می‌آیند که شماره مسیرهای قرمز رنگ به صورت sidetrip در مسیر آزمون tour شده‌اند. حال می‌بایست بر مبنای این مسیرهای آزمون، موارد آزمون را طراحی کنیم:

|  |  |  |
| --- | --- | --- |
| **شماره مسیر آزمون** | **مورد آزمون** | |
| **ورودی** | **خروجی مورد انتظار** |
| **۱** | “” | 0 |
| **۲** | “ 1” | 1 |
| **۳** | “ -t” | 0 |
| **۴** | “ t” | 0 |
| **۵** | Infeasible | |

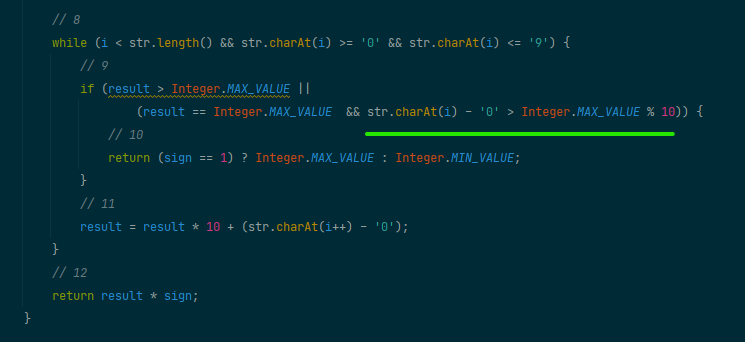
بعد از بررسی مسیر آزمون شماره ۵ روی کد برنامه، متوجه شدیم که طی مسیر مورد نظر امکان‌پذیر نیست (زیرا برای رفتن به گره شماره ۱۰، می‌بایست متغیر میانی result برابر با باشد که این یعنی حلقه ۸-۹-۱۱ می‌بایست به تعداد رقم‌های این عدد طی شود. در اینجا می‌توانیم مسیر شماره ۵ را با نیازمندی موجود تغییر دهیم و جدول بالا را بازنویسی کنیم:

|  |  |  |
| --- | --- | --- |
| **شماره مسیر آزمون** | **مورد آزمون** | |
| **ورودی** | **خروجی مورد انتظار** |
| **۱** | “” | 0 |
| **۲** | “ 1” | 1 |
| **۳** | “ -t” | 0 |
| **۴** | “ t” | 0 |
| **۵** | “ +21474836471” | 2147483647 |

بعد از تهیه موارد آزمون، کد مربوط به اجرای آزمون‌ها را در فایل StringToIntTest نوشتیم. تصویر زیر،ِ نتیجه اجرای آزمون است:



پس از بررسی مسیر آزمون ۵ متوجه می‌شویم که در یکی از خط‌های برنامه (که در مسیر مورد نظر اجرا می‌شود)، شرط نادرستی مورد ارزیابی قرار می‌گیرد. تصویر حاوی این قطعه کد در زیر آمده است:



با توجه به توضیحات موجود در سوال، عبارت شرطی که با خط سبز رنگ مشخص شده است کاملا اشتباه است، زیرا تنها در صورتی موجب خاتمه اجرای برنامه می‌گردد که مقدار متغیر میانی result برابر با Integer.MAX\_VALUE و رقم i بزرگتر از رقم آخر Integer.MAX\_VALUE (یعنی ۷) باشد. با حذف شرط فوق، می‌توان مشکل این برنامه را رفع کرد.