به نام خدا

پروژه اول درس هوش مصنوعی

استاد نیک آبادی

محمدمهدی آقاجانی

پاییز ۹۵

نحوه مدل سازی پروژه و روابط حاکم بر کلاس‌ها

در این پروژه سه package ساخته شده است . که اولی مربوط به program و بعدی algorithm و بعدی data structures میباشد که هر یک را به اختصار توضیح میدهیم :

۱- problem

این بسته شامل کلاس پدر problem میباشد که abstract است و متد های مربوط به یک مساله مانند اولین حالت و یا حالت‌های بعدی و تعیین هدف بودن و تابع شهودی در این کلاس است که باید مابقی کلاس‌های فرزند این متد ها را پیاده‌سازی کنند.

همچنین کلاس‌های فرزند Robot و Queens وجود دارند که مربوط به هر یک از این مسایل میباشند.

۲- algorithm

این بسته شامل کلاس پدر algorithm است که حاوی لیست های f , e است و همچنین متد abstract ای دارد به نام apply که همه فرزندانش باید آن را پیاده‌سازی کنند . ویژگی‌های مربوط به یک الگوریتم از قبیل تعداد گره‌های دیده شده یا تعداد گره‌های بسط داده شده و بیشترین مصرف حافظه و متد های , getter setter آن‌ها نیز در این کلاس‌ها قرار دارند .

اما هر یک از الگوریتم ها به نام خودشان دارای یک کلاس هستند که متد apply در آن‌ها پیاده سازی شده است که دارای دو حالت درختی و گرافی میباشد.برای اینکه یک الگوریتم را بر روی مساله اجرا کنیم کافی است متد apply آن را به همراه حالت اجرا (درختی یا گرافی) بر روی مساله صدا بزنیم(یکی از ورودی های این متد شی ای از جنس problem میباشد)

همچنین این بسته بر اساس جست وجو های آگاهانه یا ناآگانه دسته بندی شده است .

۳- data structures

این بسته شامل کلاس پدر node می‌باشد و کلاس‌های فرزند robot node , queens node در این بسته حضور دارند . یک node شامل اشاره گری به پدرش و همچنین هزینه دسترسی به خود است که متد

هایمربوط به آن‌ها نیز در آن‌ها پیاده‌سازی شده است . همچنین state نیز داخل node نگه داری می‌شود که برای هر مساله ساختمان داده مربوط به خود را دارد.

توضیح کلاس‌های root

۱- problem solving agent

این کلاس درواقع به عنوان حل کننده مسایل عمل میکند بدین صورت که یک مساله و یک الکوریتم میگیرد و آن الگوریتم را بر روی مساله اجرا میکند و سپس جواب بدست آمده را به کلاس response formatter میدهد تا این کلاس بتواند جواب را با فرمت درستی نمایش دهد.

۲- response formatter

مسیولیت درست و زیبا نمایش دادن جواب‌ها و اطلاعات مساله را دارد.

حال به سراغ گزارش سؤال‌ها می‌رویم :

تمرین اول : وزیر ها

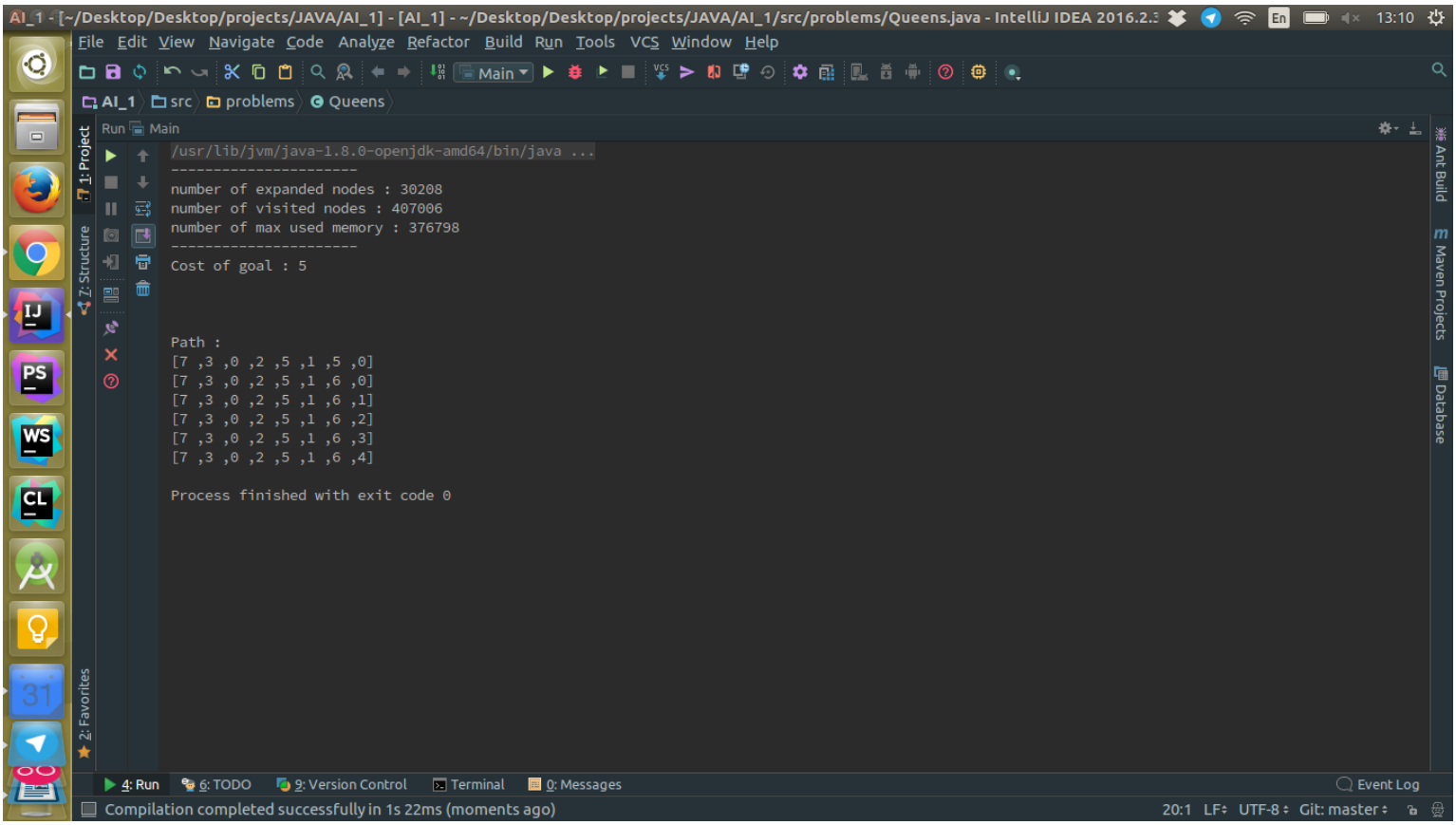
در این مساله حالت‌ها را به صورت یک آرایه هشت تایی فرض کرده‌ایم به این صورت که هر درایه مکان یک وزیر در ستون i ام را نشان میدهد.حالت اولیه به صورت <۰،۴،۱،۵،۲،۰،۳،۷> میباشد. هر حرکت به این صورت است که میتوان یک وزیر را در ستونش یک خانه به بالا و یا یک خانه به پایین آورد و تابع next state بر همین اساس لیستی از حالت‌های بعدی را به ما تحویل میدهد.تابع هدف هم چک میکند تا هیچ وزیری همدیگر را تهدید نکنند.

توضیح تابع شهودی

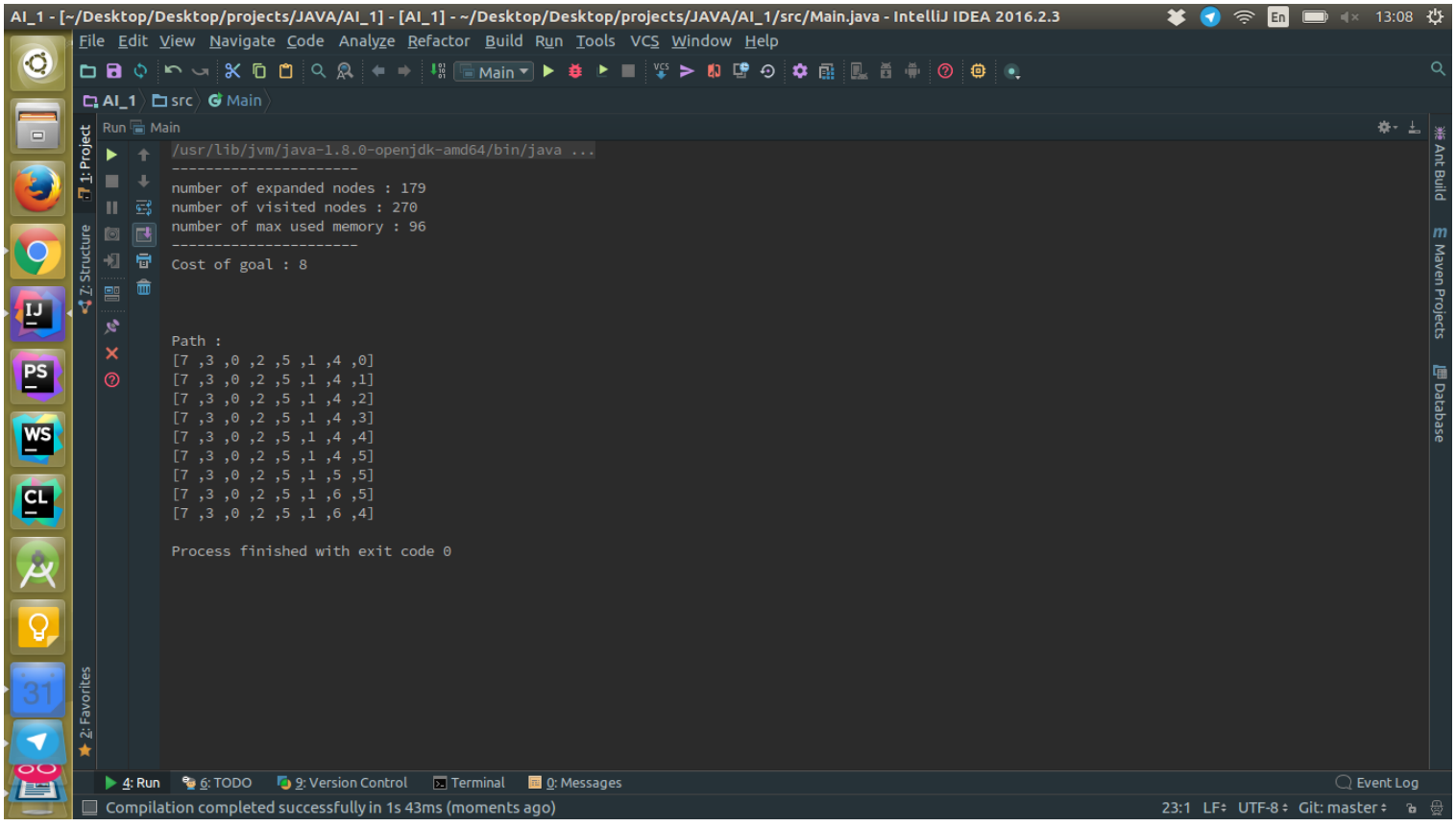
در این تمرین تابع شهودی تعداد سطر هایی است که هیچ وزیری در آن حضور ندارد . این تابع به وضوح برای حالت هدف برابر صفر است و همچنین قابل قبول نیز هست زیرا برای هر حالت از کمترین میزان هزینه رسیدن به هدف ، کمتر میباشد.در نتیجه اگر الگوریتم را به صورت درختی اجرا کنیم نتیجه بدست آمده بهینه خواهد بود.

خروجی ها

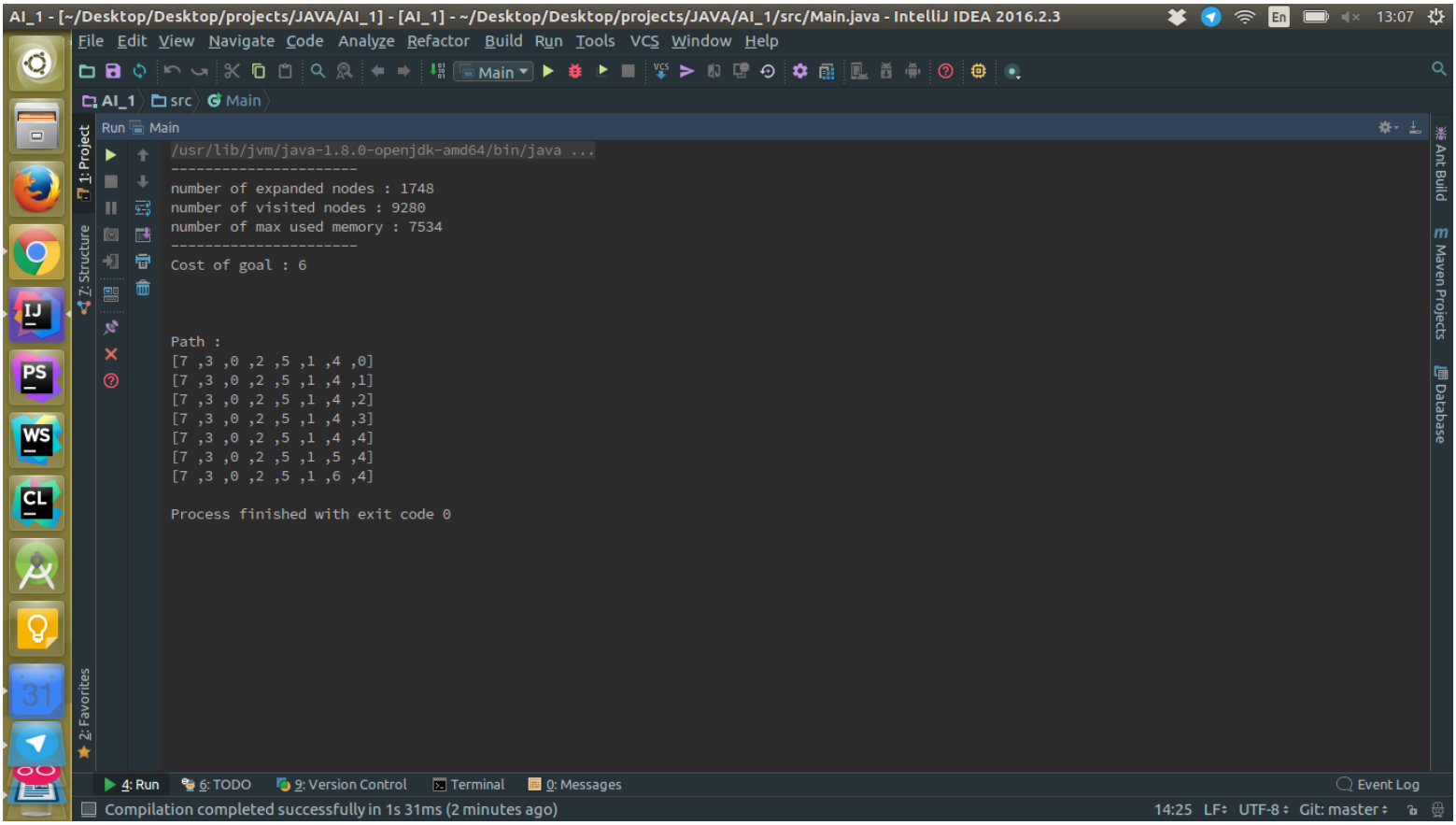
در بار اول مساله را با الگوریتم سطح اول به صورت درختی اجرا میکنیم و خروجی زیر مشاهده می شود:



سپس با الگوریتم Deepth limited serach با عمق ۸ آن را اجرا کردیم که خروجی زیر حادث شد :



و در آخر هم با الگوریتم Astar که خروجی زیر را گرفتیم :



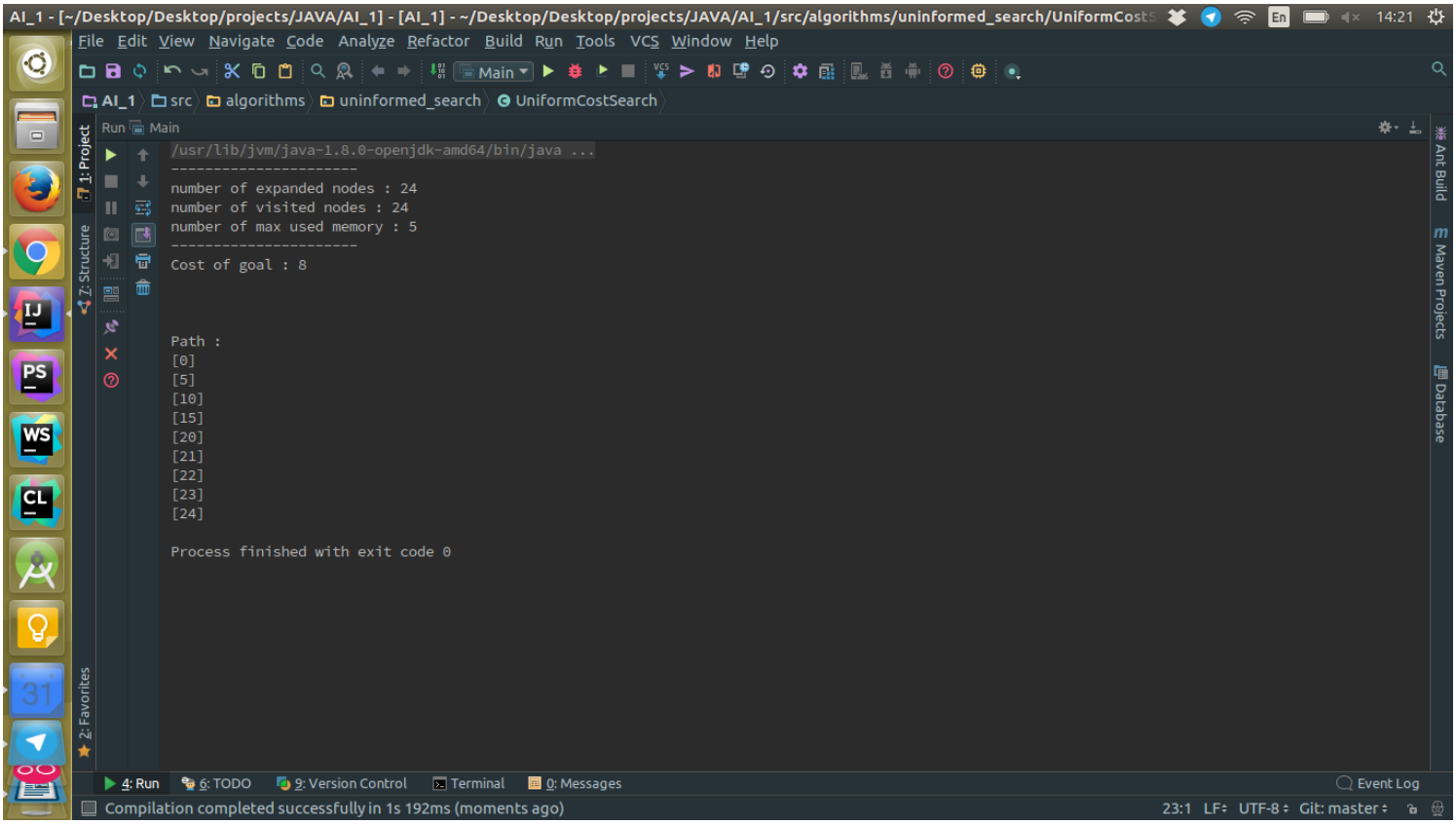
– دقت کنید که در الگوریتم اول به دلیل زمان بر بودن زیاد حالت اولیه ساده‌تر در نظر گرفته شده ولی با این حال میزان گره‌های بسط داده شده خیلی بیشتر از دو حالت بعدی ست . هم چنین در دو حالت بعدی مشاهده میکنید که الگوریتم ASTAR به صورت بهینه عمل کرده و راه کوتاهتری را انتخاب نموده است هرچن د که گره‌های بیشتر را نسبت به حالت دوم بسط داده است .

تمرین دوم : مسیریابی روبات

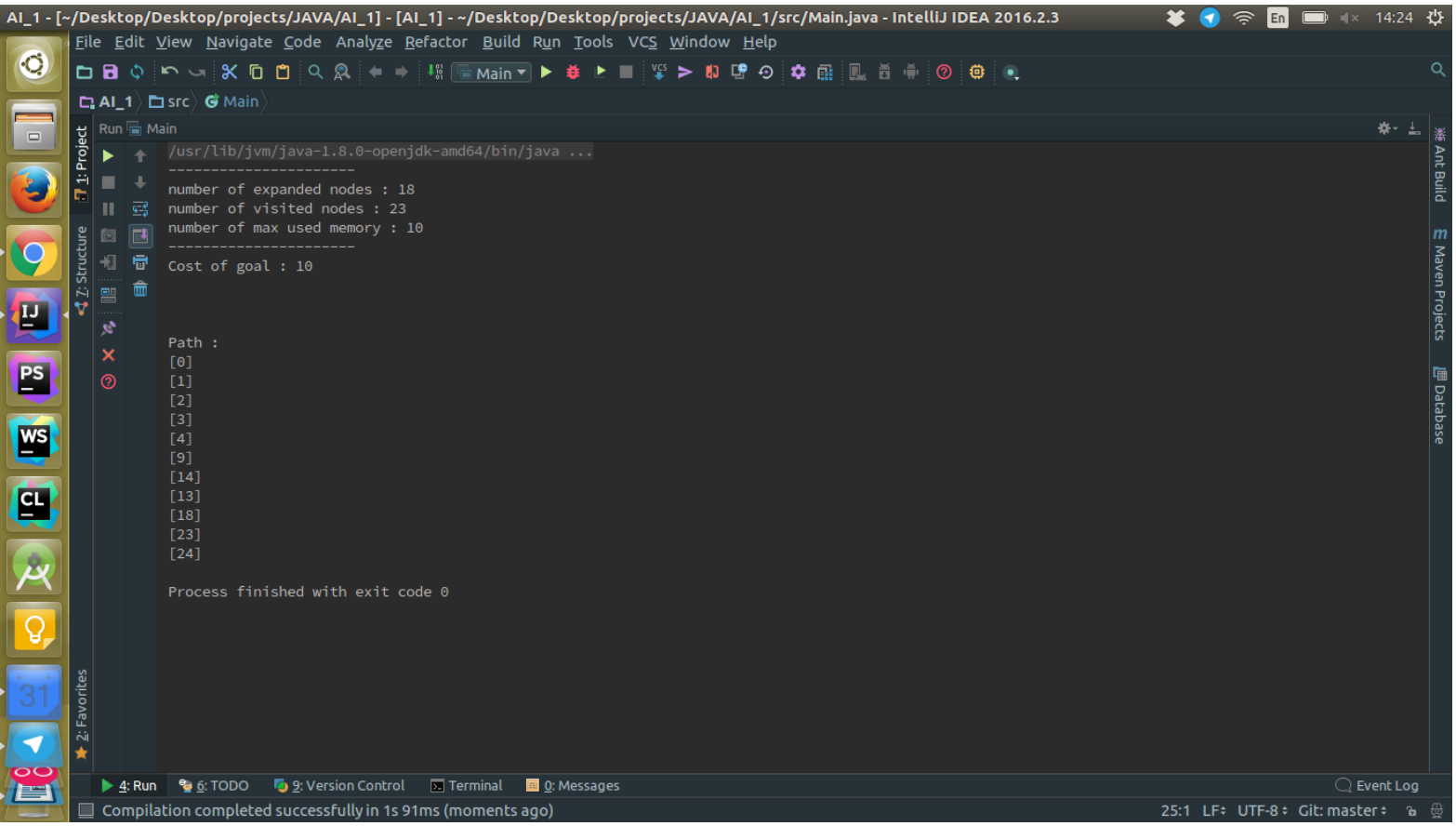
در این مساله حالت را به صورت یک عدد صحیح در نظر گرفتیم به این صورت که خانه‌های جدول شماره گذاری شده‌اند و اولین خانه شماره ۰ و آخرین خانه (در حالت ۵*۵) برابر ۲۴ است پس هر حالت بیانگر حضور ربات در یکی از خانه‌های جدول میباشد.

حال به سراغ خروجی ها می‌رویم :

در ابتدا جست‌وجوی هزینه یکنواخت را اجرا میکنیم :

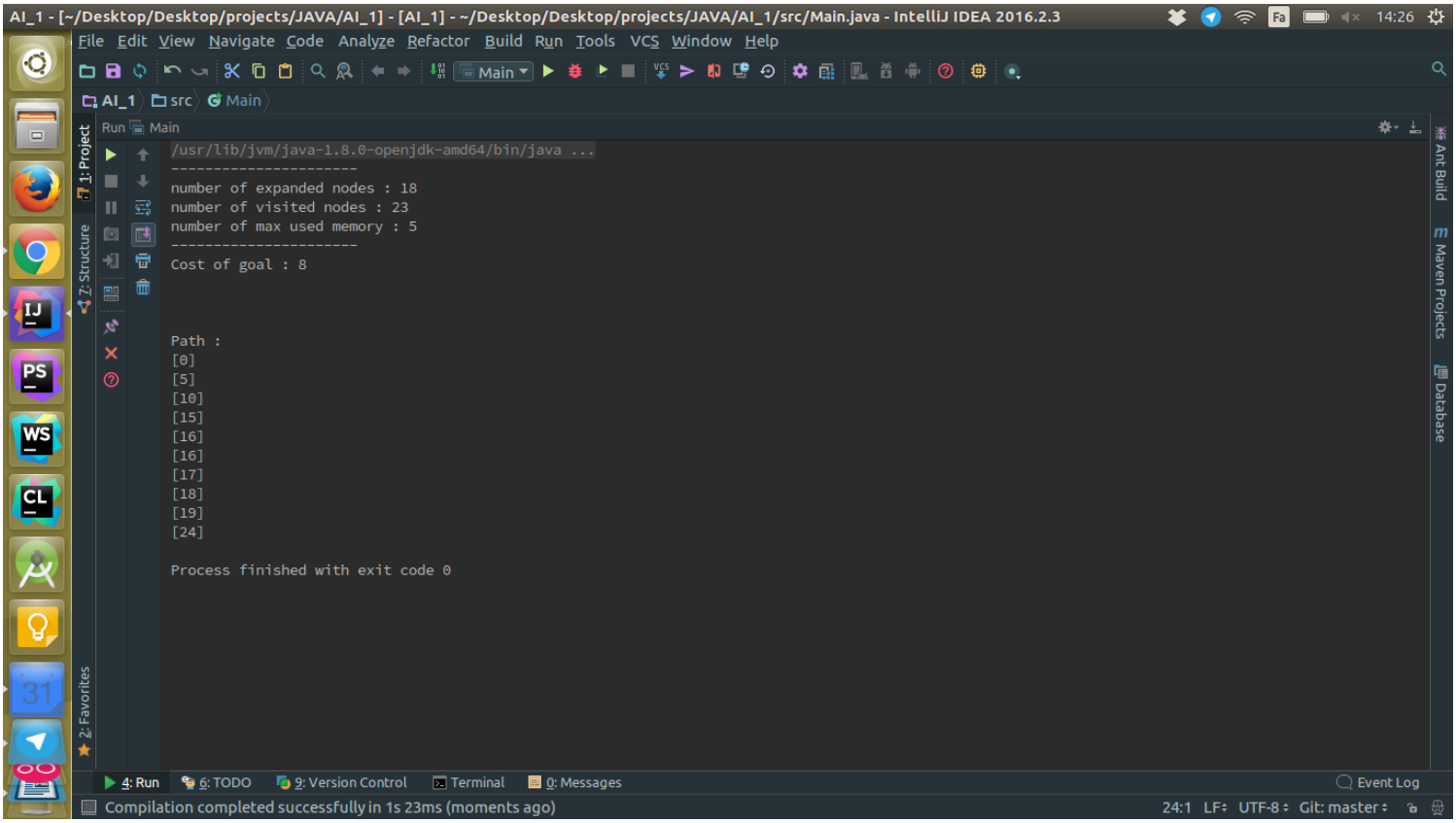


سپس عمق اول را اجرا میکنیم :

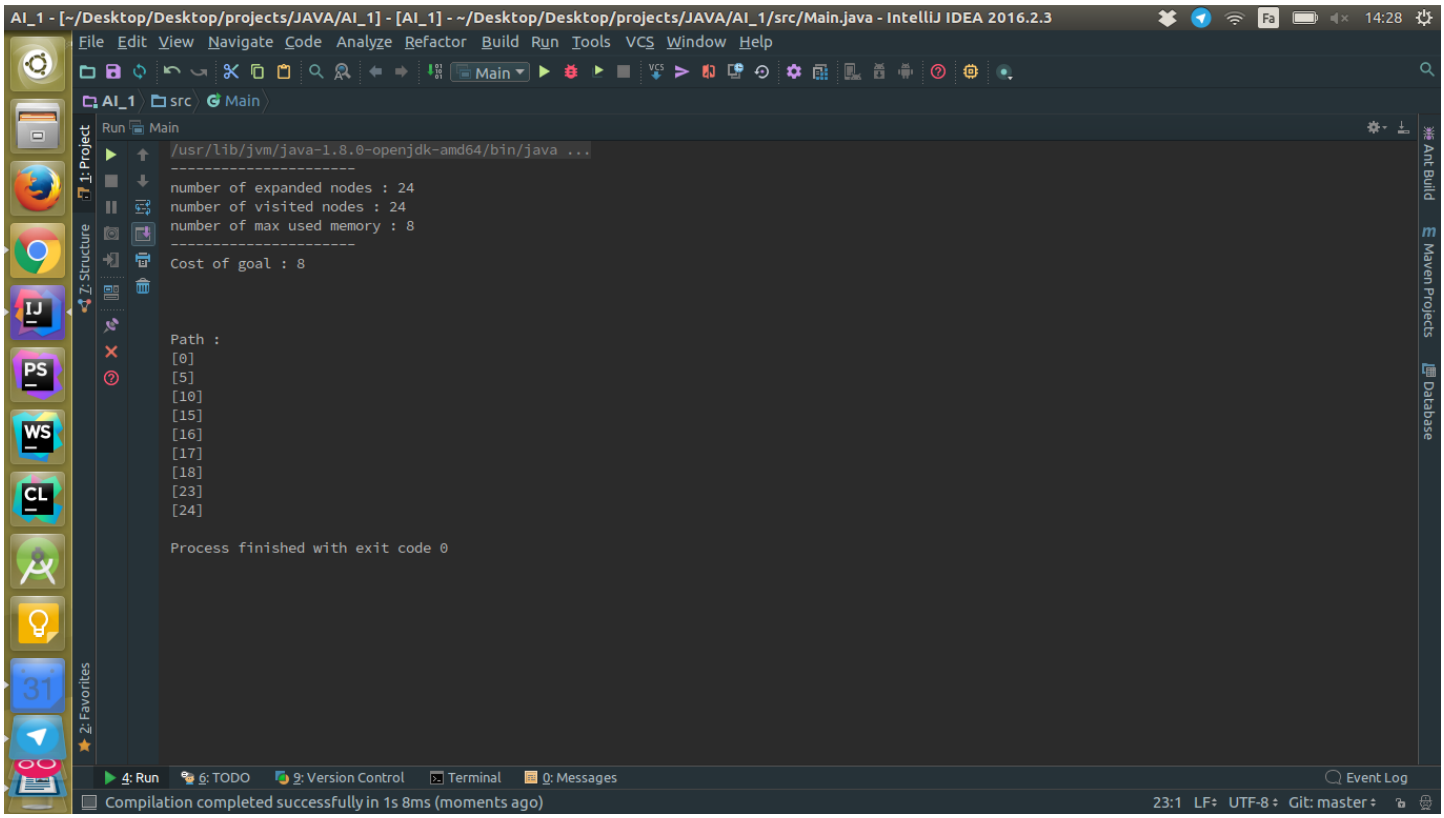


و حال جست‌وجوی دوطرفه را اجرا مینماییم :

همانطور که میبیند در گره [۱۶] دو طرف به یکدیگر پیوسته اند :



و در نهایت ASTAR را اجرا مینماییم :



توضیح :همانگونه که میبینید مطابق انتظار جست‌وجوی عمق اول بهینه نیست و مسیری که حداقل در ۸ گام میتوان پیمود را در ۱۰ گام میپیمایداز طرفی نسبت به Astart و uniform cost search گره‌های کمتری را بسط میدهد. همچنین جست‌وجوی دوطرفه که از دو طرف به صورت سطح اول جست‌وجو میکند به صورت گرافی اجرا شده در نتیجه گره زیادی را بسط نمیدهد.

مسیر قابل پیمایش نیز برای الگوریتم های مختلف متفاوت است که در تصاویر قابل مشاهده است