

الگوریتم پیشرفته

تمرین ۲

نیلوفر وفا

۴۰۱۴۲۲۱۹۶

1.1

رویکرد غیر هوشمندانه brute-force به این مسئله به این صورت است که همه‌ی حالت‌های ممکن را امتحان کنیم. یعنی برای هر مهره در ابتدای ردیف، آن را انتخاب کنیم و سپس مسئله را به صورت بازگشتی بر روی ردیف باقی‌مانده حل کنیم. این روش بهینه نیست زیرا تعداد حالت‌ها به صورت نمایی با افزایش تعداد مهره‌ها افزایش می‌یابد و زمان اجرای آن بسیار زیاد می‌شود.

1.2 یک استراتژی هوشمندانه برای پاسخ بهینه به این مسئله، استفاده از الگوریتم دینامیک است. می‌توان از یک جدول دویبعی برای ذخیره‌سازی امتیازهای حاصل شده در هر بازه از ردیف استفاده کرد. در این الگوریتم، با شروع از ابتدای ردیف، به صورت بازگشتی امتیازهای حاصل شده را محاسبه می‌کنیم. برای هر حالت، می‌توانیم یکی از دو انتخاب را داشته باشیم: انتخاب اولین مهره یا انتخاب آخرین مهره. با انتخاب هر کدام از آنها، امتیاز به دست آمده و مجموع امتیازهای بدست آمده در ادامه ردیف را به عنوان ورودی به بازگشت استفاده می‌کنیم. سپس از بین دو حالت بالا، حالتی را انتخاب می‌کنیم که امتیاز بیشتری دارد. در نهایت، با ادامه این فرآیند تا انتهای ردیف، امتیاز حاصل شده و همچنین دنباله حرکات مربوطه به دست می‌آید.

1.3 پیچیدگی زمانی الگوریتم دینامیک ارائه شده برابر با $O(n^2)$ است. چرا که برای هر حالت در ردیف، حداکثر دو حالت (انتخاب اولین مهره و انتخاب آخرین مهره) را در نظر می‌گیریم و امتیازهای حاصل شده را در جدول دویبعی ذخیره می‌کنیم. تعداد کل خانه‌ها در جدول برابر با n^2 است.

۲-

حل مسئله شما مربوط به انتخاب بهینه ترین مکان‌های توقف در طول مسیر است که به صورت حریصانه قابل حل است. الگوریتم حریصانه ارائه شده در ادامه قابل استفاده است:

۱- خواندن ورودی:

مقدار n را برابر با حداکثر فاصله که می‌توانید با پر شدن باک ماشین طی کنید قرار دهید.

خواندن مختصات جایگاه‌های سوخت از نقشه راه.

۲- محاسبه فواصل بین جایگاه‌های سوخت:

محاسبه فاصله بین هر جفت جایگاه سوخت با استفاده از فرمول فاصله دو نقطه در فضای دوبعدی (مانند فاصله اقلیدسی).

۳- الگوریتم حریصانه:

مقدار پیشفرض برای تعداد توقف‌ها (count) را صفر قرار دهید.

ایجاد یک لیست خالی برای ذخیره مکان‌های توقف (stops).

تا زمانی که باک ماشین پر نشده است:

پیدا کردن نزدیک‌ترین جایگاه سوخت (مکان جایگاه) به موقعیت فعلی.

اگر فاصله جایگاه سوخت به موقعیت فعلی از باقی فاصله‌ی باک کمتر یا مساوی n است:

اضافه کردن مکان جایگاه به لیست توقف‌ها.

افزایش مقدار count به ازای توقف جدید.

کاهش مقدار باک با توجه به فاصله بین موقعیت فعلی و مکان جایگاه.

تغییر موقعیت فعلی به مکان جایگاه.

در غیر اینصورت، قطعه کد را تمام کنید.

۴- چاپ خروجی:

چاپ تعداد کل توقف‌ها. (count)

چاپ مختصات مکان‌های توقف به ترتیب آنها در لیست توقف‌ها.

الگوریتم حریصانه بهینه‌ترین مکان‌های توقف را با توجه به فاصله بین جایگاه‌های سوخت و مکان فعلی شما انتخاب می‌کند. با اجرای این الگوریتم، می‌توانید توقف‌هایی را پیدا کنید که فاصله آنها تا موقعیت فعلی شما کمتر یا مساوی باقیمانده‌ی باک ماشین (n) باشد، بدین ترتیب تعداد توقف‌ها را به حداقل ممکن می‌رسانید و تعداد دفعات پر کردن باک را کاهش می‌دهید.

با اجرای الگوریتم حریصانه، می‌توانید به صورت کارا و بهینه مسیر خود را برنامه‌ریزی کنید تا توقف‌های لازم را در مکان‌های مناسب داشته باشید و تعداد توقف‌ها و مسافت طی شده را به حداقل برسانید.

در این حالت، طول جاده‌ها به صورت توانی از ۲ تعریف شده است، که اگر طول جاده ۸ کیلومتر باشد، برابر است با ۳ به توان ۲ کیلومتر. بر اساس این توضیحات، اسکریپت پایتون زیر را برای محاسبه مجموع حداقل فواصل بین هر جفت شهر و چاپ خروجی به صورت دودویی ارائه می‌دهیم.

با توجه به ، خروجی باید 100100 باشد که معادل عدد دهدهی 68 است.

در این اسکریپت، از الگوریتم فلویید-وارشال برای پیدا کردن حداقل فواصل بین تمام جفت شهرها استفاده شده است. توسط این الگوریتم، فواصل کمینه بین تمام جفت شهرها به روزرسانی می‌شوند و در نهایت مجموع این فواصل کمینه محاسبه می‌شود.