تمرین ۲
الگوریتم‌های پیشرفته - دکتر فراهانی
مشکات احمدی

بسمه تعالی

سوال ۱)

پاسخ سوالات و کد در نوتبوک آورده شده است.

سوال ۲)

راه حل پیشنهادی این مساله یک الگوریتم حریصانه است به این صورت که با مراجعه به نقشه بین تبریز و زاهدان ابتدا دورترین جایگاه سوخت در مسیر که در حدفاصل n کیلومتریست را شناسایی می‌کنیم، و بعد از پرکردن باک در این شهر، شهر بعدی را با تکرار همین روال پیدا می‌کنیم تا زمانی که به مقصدمان زاهدان برسیم.

برای اثبات بهینه بودن این راه حل فرض کنید که راه حل حریصانه توقف در ایستگاههای $a_1, a_2, \cdots, a_k$ را پیشنهاد دهد.

نقطه k+1 همان زاهدان باشد. فرض کنید مسیر دیگری وجود دارد که $b_1, b_2, \cdots$ است و مسیر بهینه‌تری از راه حل حریصانه ماست. می‌خواهیم نشان دهیم که به ازای هر i در $1 \leq i \leq k$ هیچ کدام از b_iها از a_i متناظر به مقصد نزدیکتر نیست. بنابرین به این طریق اثبات می‌کنیم تعداد توقفات مسیر حریصانه بهینه است.

از طریق استقرا ثابت می‌کنیم.

۱) به ازای $i = 1$ بدیهی است که a_1 یا بهتر یا دقیقا همان b_1 است. چرا که در اولین گام بیشترین مسیر ممکن تا مقصد را آمده ایم.

۲) فرض کنیم به ازای i بیشتر از یک این فرض برقرار باشد که مسیر $a_1, a_2, \cdots, a_i$ از مسیر $b_1, b_2, \cdots, b_i$ به مقصد نزدیکتر شده باشد. در این صورت در مرحله بعدی b_{i+1} حداکثر به اندازه a_{i+1} یا کمتر از آن می‌تواند به مقصد نزدیک شده باشد چرا که a_i به مقصد از b_i نزدیکتر است و از a_i نمی توان به مقصد بیش از a_{i+1} جلو رفت بنابرین با مفروض گرفتن بهینه بودن مساله برای i به بهینه بودن مسیر برای i+1 رسیدیم که a_{i+1} هم از b_{i+1} به مقصد نزدیکتر است.

با توجه به صحت پایه استقرا و گام استقرایی، به ازای تمام iهای بزرگتر از ۱ این مسیر حریصانه مسیر بهینه با کمترین توقفات تا زاهدان است.

سوال ۳)

برای پیدا کردن مجموع کمینه فاصله‌ها ابتدا کمینه فاصله‌ها را از الگوریتم فلوید وارشال محاسبه می‌کنیم. از آنجا که یالها مثبت هستند (توان دو) می توانیم از این الگوریتم برنامه نویسی پویا استفاده کنیم. به این صورت که برای ماتریس D که ماتریس فاصله است و Dk نشانگر، مرحله k از محاسبه است (که k) گره میانی بین دو گره است:

$D_k [i , j] = \min (D_{k-1} [i , j], D_{k-1} [i, k] + D_{k-1} [k , j])$

تمرین ۲

الگوریتم‌های پیشرفته - دکتر فراهانی

مشکات احمدی

یالهای تعریف نشده باید مقدار بی‌نهایت بگیرند و از این رو مقدار ماکزیمم که 2* 10^5 را به آنها می‌دهیم.

از آنجا که مقادیر یالها به توان دو هستند و نمایش باینری دارند برای جمع و مقایسه آنها می‌توانیم از عملگرهای باینری استفاده کنیم که از این جهت توابع min_binary_string و add_binary_strings برای مینیمم گیری و جمع دو رشته باینری ایجاد شد تا در برنامه نویسی پویا از اینها استفاده شود.

کد و خروجی در نوتبوک قرار داده شده است.