

هو العلم



طراحی و تحلیل الگوریتمها

نیمسال دوم سال تحصیلی ۱۴۰۱ - ۱۴۰۰

تمرینات برنامه نویسی ۱

مریم رضائی

۱. می‌خواهیم مبتنی بر الگوریتم هافمن، برنامه‌ای برای کدگذاری (فشرده‌سازی) متون و برنامه‌ای مستقل نیز برای کدگشایی (منبسط‌سازی) متون فشرده بنویسیم. فرض کنید متونی که قرار است فشرده شوند با کدهای آسکی^۱ در رایانه ذخیره شده باشد.

برنامه‌ی کدگذار باید یک فایل متنی را به عنوان ورودی بخواند و یک فایل متنی را به عنوان خروجی بنویسد. به بیان دقیق‌تر، برنامه فشرده‌ساز باید

❖ برای بار اول متن ورودی را بخواند و فراوانی (تعداد دفعات تکرار) هر یک از نویسه‌های (کاراکترهای) آن را تعیین کند؛

❖ درخت کدگذاری هافمن را بسازد؛

❖ جدول حاوی کلمات کد (رشته‌های بیتی) هر یک از کاراکترهای متن را بسازد.

❖ درخت هافمن را به شکل یک رشته بیتی (در ابتدای فایل خروجی) بنویسد.

❖ تعداد کاراکترهای متن ورودی را به شکل یک رشته بیتی (بعد از رشته بیتی درخت) بنویسد.

❖ برای بار دوم متن ورودی را بخواند و با استفاده از جدول کلمات کد، به جای هر کاراکتر در متن، رشته بیتی معادل با آن را بگذارد تا رشته بیتی فشرده‌ی معادل با متن اصلی را به دست آورد. و این رشته بیتی را (بعد از رشته بیتی معادل با تعداد کاراکترهای متن ورودی) در فایل خروجی بنویسد.

❖ بعد از فشرده کردن متن اصلی و ذخیره کردن آن در فایل خروجی، هم متن اصلی و هم متن فشرده و هم نسبت فشرده‌سازی (تعداد بیت‌های نمایشگر متن فشرده به تعداد بیت‌های نمایشگر متن اصلی) را در پنجره خروجی چاپ کند.

برنامه کدگشا باید یک فایل متنی را که قبلاً با برنامه کدگذار فشرده شده است، به عنوان ورودی بخواند و با پردازش آن، فایل متنی اصلی را بازیابی کند. به بیان دقیق‌تر، برنامه منبسط‌ساز باید

❖ درخت هافمن را (که به شکل یک رشته بیتی در ابتدای متن فایل ورودی قرار گرفته است) بسازد.

❖ با خواندن قسمتی دیگر از متن، تعداد کاراکترهایی را که باید آنها را کدگشایی کند تعیین کند.

❖ با استفاده از درخت، قسمت باقیمانده متن را (که در واقع، همان متن فشرده است) منبسط کند و آن را در

¹ ASCII

یک فایل جدید ذخیره کند.

❖ بعد از منبسط کردن متن اصلی و ذخیره کردن آن در یک فایل، هم متن فشرده و هم متن اصلی را در پنجره خروجی چاپ کند.

جواب:

توصیف کلی کار برنامه‌ها: در دو برنامه مجزا، کدگذاری و کدگشایی را می‌نویسیم به طوری که برنامه کدگشا نام یک فایل متنی در پوشه برنامه را گرفته و آن را به صورت یک رشته می‌خواند و در خروجی فایلی در پوشه می‌سازد یا تغییر می‌دهد که دارای کد شده‌ی درخت، طول متن و کد شده‌ی متن در سه سطر مجراست. برای این کار، برنامه از توابع متفاوت استفاده می‌کند که هر کدام در برنامه با کامنت‌ها توضیح داده شده‌اند. در دنباله آن، برنامه کد گشا نام یک فایل متنی در همان پوشه که حاوی سه سطر کد درخت و کد متن است را گرفته و ابتدا درخت را می‌سازد و سپس با استفاده از آن، متن را کد گشایی می‌کند تا در نهایت آن را در فایلی دیگر ذخیره کند.

مراجع:

- <https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3>
- <https://www.geeksforgeeks.org/python-frequency-of-each-character-in-string>

۲. فرض کنید می‌خواهیم در یک شبکه رایانه‌ای، از رایانه‌ای خاص به عنوان رایانه مبدأ، جریانی پیوسته از داده‌ها را با حداکثر سرعت ممکن به رایانه‌ای دیگر به عنوان رایانه مقصد، منتقل کنیم. برای آنکه بتوانیم میزان بیشتری از داده‌ها را در واحد زمان ارسال کنیم، می‌توانیم داده‌ها را به بسته‌هایی تقسیم کنیم و آن بسته‌ها را از طریق مسیرهای مختلف از مبدأ به مقصد بفرستیم. از طرف دیگر، برای ارسال داده‌ها هم با این قید مواجه‌ایم که نمی‌توان جریان داده‌ها را با سرعتی بیشتر از یک مقدار مشخص از هر خط ارتباطی بین دو رایانه عبور داد و و هم با این قید مواجه‌ایم که نمی‌توان جریان داده‌ها را با سرعتی بیشتر از یک مقدار مشخص از هر رایانه میانی (هر مسیریاب) عبور داد.

این شبکه رایانه‌ای را می‌توان با یک گراف جهت‌دار وزن‌دار نمایش داد: هر رایانه عضو شبکه را می‌توان رأسی از رئوس گراف در نظر گرفت و هر خط ارتباطی یک طرفه بین دو رایانه را می‌توان یالی از یال‌های جهت‌دار گراف دانست. پهنای باند هر خط ارتباطی (که حداکثر تعداد بایت‌هایی است که می‌توان در یک ثانیه از آن خط عبور داد) در شبکه، وزن یک یال جهت‌دار در گراف را مشخص می‌کند و پهنای باند هر رایانه در شبکه، وزن یک رأس در گراف را مشخص می‌کند.

برنامه‌ای برای حل حالت کلی این مسأله بنویسید. این برنامه باید گراف جهت‌دار $G = (V, E)$ و دو رأس مبدأ s و مقصد t را بگیرد و حداکثر میزان داده‌هایی را که می‌توان در واحد زمان از رایانه s به رایانه t منتقل کرد، و همچنین مسیرهای ارسال بسته‌های داده را تعیین کند.

جواب:

توصیف راه: در اصل سوال همان مسئله جریان بیشینه است با این تفاوت که رئوس وزن داشته و مسیرها را نیز باید خروجی دهیم. به این علت در پیش روی و یافتن راه، کافیسیت برای هر راه یافت شده با پیمایش سطحی، مسیر را در آرایه‌ای ذخیره کنیم. همچنین برای حل مشکل وزن رأس، می‌توانیم رئوس را در اصل یالی دیگر در نظر گرفته (یعنی یال‌های اصلی جایگزین شده و یال‌های گراف جدید دو برابر شوند) و برای این گراف جدید مسئله جریان بیشینه را حل کنیم. برای خروجی، جریان یکسان خواهد بود و فقط کافیسیت رئوس مسیر یافت شده را به رئوس گراف اصلی تبدیل کنیم که برای این کار می‌دانیم رئوس جدید دو برابر رئوس قبلی هستند و می‌توان رأس معادل را با محاسبات یافت.

مشکل برنامه: برنامه در بخش محاسبه جریان بیشینه گراف تبدیل شده به مشکلی برخورد و با در نظر گرفتن یال جایگزینی به عنوان یک مسیر محدود، از آن گذشته و از جریان باقی‌مانده کم می‌کند و بدین طور جلوی پیشرفت خود از یال اول را می‌گیرد. متأسفانه فرصت فکر و تصحیح این ایراد را نداشتیم.

مراجع:

- <https://www.geeksforgeeks.org/ford-fulkerson-algorithm-for-maximum-flow-problem>