عنوان: پیاده‌سازی کدگذاری هافمن
نویسندگان: امیرحسین صبری، کیمیا کیوانلو
:مقدمه
پیاده‌سازی کدگذاری هافمن یک برنامه پایتون است که عملیات فشرده‌سازی و بازگشایی داده با استفاده از الگوریتم کدگذاری هافمن را نشان می‌دهد. این برنامه امکانات مختلفی ارائه می‌دهد از جمله ساختن درخت هافمن، کدگذاری و بازگشایی داده، نمایش توزیع فراوانی، مقایسه و تحلیل کارآیی فشرده‌سازی ASCII با کدهای.
:وابستگی‌ها
heapq: این ماژول یک پیاده‌سازی از الگوریتم صف دودویی فراهم می‌کند که برای ساخت درخت هافمن استفاده می‌شود.
collections.defaultdict: این ماژول یک کلاس defaultdict را ارائه می‌دهد که به مقادیر پیش‌فرض برای کلیدهایی که در یک دیکشنری تعیین نشده‌اند، امکان می‌دهد.
tkinter: این ماژول کتابخانه Tkinter را برای ایجاد رابط کاربری گرافیکی برای نمایش درخت هافمن فراهم می‌کند.
colorama: این ماژول امکان پشتیبانی از خروجی رنگی ترمینال را فراهم می‌کند.
art: این ماژول قلم‌های ASCII art را برای نمایش عنوان برنامه فراهم می‌کند.
:توابع
draw_tree(node, x, y, dx): درخت هافمن را بر روی نقاشی Tkinter نمایش می‌دهد.
printCodes(root, code_str): کدهای هافمن را برای هر کاراکتر در درخت چاپ می‌کند.

کدهای هافمن را برای هر کاراکتر در :storeCodes(root, code_str)
.ذخیره می‌کند codes دیکشنری
درخت هافمن را ساخته و کدهای هافمن را ذخیره :HuffmanCodes(size)
.می‌کند
فراوانی هر کاراکتر را در رشته ورودی محاسبه :calcFreq(input_str, n)
.می‌کند
یک رشته کدگذاری شده با استفاده از درخت :decode_file(root, s)
.هافمن ارائه شده را بازگشایی می‌کند
ASCII کدهای هافمن را با کدهای :compare_with_ascii(input_str)
.کاراکترها در رشته ورودی مقایسه می‌کند
توزیع فراوانی :visualize_frequency_distribution(input_str)
کاراکترها در رشته ورودی را محاسبه کرده و با استفاده از نمودار نمایش
.می‌دهد
میزان کارآیی فشرده‌سازی :efficiency_comparison(input_str)
.کدگذاری هافمن را با تکنیک‌های فشرده‌سازی دیگر مقایسه می‌کند
یک فایل را با استفاده از کدگذاری هافمن :compress_file(file_path)
.فشرده می‌کند و داده‌های فشرده شده را در یک فایل جدید ذخیره می‌کند
نسبت :calculate_compression_ratio(input_str, encoded_str)
.فشرده‌سازی بین رشته اصلی و رشته کدگذاری شده را محاسبه می‌کند
:پیچیدگی زمانی
تعداد گره‌ها در درخت N که در آن ،draw_tree(node, x, y, dx): O(N)
.است
تعداد گره‌ها در درخت N که در آن ،printCodes(root, code_str): O(N)
.است
تعداد گره‌ها در درخت N که در آن ،storeCodes(root, code_str): O(N)
.است

تعداد کاراکترهای N که در آن ،HuffmanCodes(size): O(N log N)
.منحصر به فرد در رشته ورودی است
.طول رشته ورودی است n که در آن ،calcFreq(input_str, n): O(n)
طول رشته کدگذاری شده m که در آن ،decode_file(root, s): O(m)
.است
طول رشته ورودی n که در آن ،compare_with_ascii(input_str): O(n)
.است
که در آن ،visualize_frequency_distribution(input_str): O(n log n)
.طول رشته ورودی است n
.برای محاسبه کدهای هافمن efficiency_comparison(input_str): O(n)
اندازه فایل ورودی n که در آن ،compress_file(file_path): O(n log n)
.است
calculate_compression_ratio(input_str, encoded_str): O(1).
به طور خلاصه، پیچیدگی زمانی کل برنامه بستگی به تابع خاصی دارد
O(n log n) تا O(n) که در حال اجرا است و می‌تواند در بازه زمانی
.باشد
:استفاده
.را تغییر دهید input_str برای تغییر رشته ورودی، متغیر
اسکریپت را اجرا کنید تا نتایج را در کنسول و رابط کاربری گرافیکی
.مشاهده کنید
:محدودیت‌ها و ارتقاهای آینده
.پیاده‌سازی کنونی تنها از ورودی‌های متنی پشتیبانی می‌کند
رابط کاربری گرافیکی ساده است و برای درخت‌های بزرگ ممکن است
.مقیاس‌پذیری نداشته باشد
این برنامه می‌تواند برای پردازش فایل‌های دودویی و ارائه تکنیک‌های
.فشرده‌سازی پیشرفته‌تر ارتقاء یابد

هندل کردن خطا و اعتبارسنجی می‌تواند برای مواجهه با حالات خاص و خطاهای ورودی بهبود یابد.

نتیجه‌گیری:

پیاده‌سازی کدگذاری هافمن یک نمونه عملی از کدگذاری هافمن برای فشرده‌سازی داده‌ها فراهم می‌کند. این برنامه ساخت درخت هافمن، نمایش توزیع ASCII، کدگذاری و بازگشایی داده، مقایسه با کدهای فراوانی و مقایسه کارایی با تکنیک‌های فشرده‌سازی دیگر را نشان می‌دهد. این برنامه می‌تواند بر اساس نیازها و مورد استفاده خاص سفارشی شده و تغییر یابد.

1.