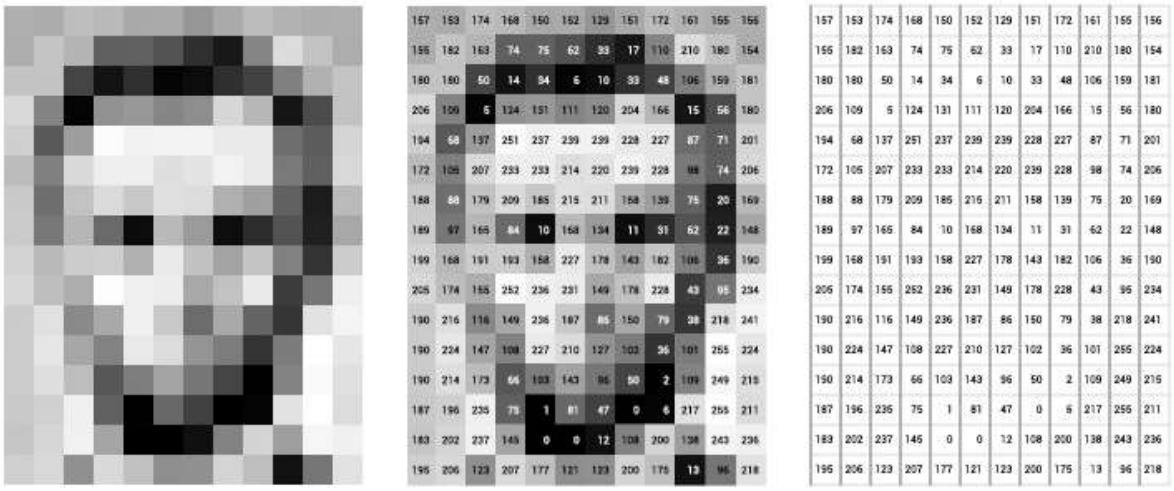
پروژه پایانی طراحی الگوریتم

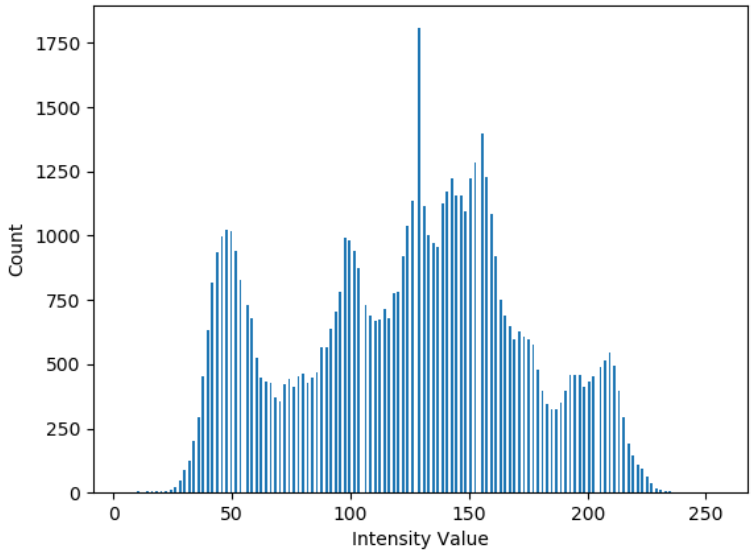
در این پروژه می‌خواهیم برنامه‌ای بنویسیم که قادر باشد یک تصویر را دریافت کرده و آن را با استفاده از کد هافمن فشرده نماید. همچنین باید قادر باشد، کد فشرده‌شده یک تصویر (کد هافمن) را دریافت و تصویر معادل با آن را بازگرداند.

در اینجای میخواهیم از تصاویر grayscale یا همان تصاویر خاکستری (غیر رنگی) استفاده نماییم. هر تصویر، از تعدادی پیکسل تشکیل شده است که مقادیر روشنایی آنها بین 0 تا 255 تغییر می‌کند. 0 نماینده سیاه و 256 نماینده سفید است. بقیه مقادیر مابین آنها سطوح مختلف رنگ خاکستری است که هرچه کمتر باشد تیره‌تر است. به مثال زیر دقت کنید.



بنابراین هر تصویر معادل یک آرایه از مقادیر 0 تا 255 است.

قبلا در کلاس با شیوه ساخت درخت هافن و در نهایت کد کردن یک متن با استفاده از کد هافمن آشنا شده‌اید. روال کار در اینجا هم همان است. فقط ما در یک تصویر بجای کلمات، سطوح روشنایی را داریم و تعداد پیکسلهایی از یک تصویر که دارای یک روشنایی هستند، تعداد دفعات رخداد (فراوانی) آن سطح از دوشنایی را مشخص می‌کنند. به عنوان مثال هیستوگرام زیر فراوانی سطوح رنگی در یک تصویر را نشان می‌دهد. همانطور که در هیستوگرام روشن است، بعضی از سطوح روشنایی اصلا در تصویر مربوطه ظاه نشده‌اند.



برای تولید کد هافمن نیاز به یک صف اولویت داریم. برای ساخت یک صف اولویت یکی از بهترین راه‌ها استفاده از هرم (heap) است. در این پروژه می‌خواهیم از همین راه استفاده کنیم. یعنی شما باید یک صف اولویت بنویسید، که ابتدا حاوی تمامی عناصر است (build). فرض کنید هر یک از اینها یک نود از یک درخت باینری است که لینکهای چپ و راست آن هردو null هستند. سپس طبق روال کد هافمن دو عنصری را که دارای کمترین فراوانی هستند را از صف خارج نمایید (delete). این دو را با هم جمع کرده، یک نود جدید میسازید که مقدارش معادل مجموع دو نود انتخابی است و نود اول به عنوان فرزند سمت راست و دیگری به عنوان فرزند سمت چپ آن قرار گرفته‌اند. این نود جدید به صف اضافه می‌شود (insert). و همین روال آنقدر تکرار می‌شود تا تنها یک نود در صف باقی بماند و این نود را که ریشه درخت هافمن نهایی است به عنوان جواب از صف خارج کرده و با استفاده از آن کد مربوط به هر عنصر را تولید می‌کنیم.

فایل تست:

برای تست این برنامه ابتدا یک آرایه 8 در 8 بسازید که به صورت تصادفی از مقادیر 0، 50، 100، 150، 200 و 250 (برای محدود کردن مقادیر) پر شده است. دقت کنید که در آرایه شما تعداد تکرار همه این مقادیر یکسان نباشد. این آرایه را در یک فایل ذخیره کرده و کد هافمن معادل آن را به صورت دستی محاسبه کرده و در یک گزارش کنار کار خود تحویل دهید. سپس همین آرایه را به برنامه خود داده و نشان دهید که برنامه شما کد را به درستی تولید می‌کند.

آنچه باید تحویل دهید:

- ✓ گزارشی حاوی:
  - تشریح محاسبه کد هافمن تصویر تصادفی مذکور در بالا
  - محاسبه میزان ذخیره‌سازی انجام‌شده در تصویر
- ✓ کد کامل پروژه

نکاتی که در پیاده‌سازی این پروژه باید در نظر داشته باشید:

- ✓ صف اولویت هم باید تماما توسط خود شما پیاده‌سازی شده و حتما یک هرم باشد.
- ✓ باید بتواند زمان اجرای ساخت درخت هافمن و تولید کد هافمن را محاسبه نماید (بر اساس واحد زمان)
- ✓ برنامه باید قادر باشدکد هافمن را دریافت کرده و تصویر را بازسازی کند.
- ✓ برنامه باید قادر باشد هر تصویری را به صورت یک فایل دریافت کرده و کد هافمن را برای آن محاسبه نماید.
- ✓ زبان مورد استفاده محدودیتی ندارد.

عوامل موثر در نمره پروژه:

- ✓ بدون خطا اجرا شدن برنامه. (وتویی: سایر نمرات در صورت برقراری این شرط، به پروژه مربوطه تعلق خواهد گرفت)
- ✓ خوانایی کد (30 نمره):
  - استفاده از comment، نامگذاری مناسب متغیرها، رعایت indent و فاصله‌گذاری‌های مناسب
- ✓ عملکرد درست برنامه (40 نمره)
- ✓ سرعت مناسب اجرا (10 نمره)
- ✓ محاسبه زمان اجرا (10 نمره)
- ✓ محاسبه درست میزان فشرده‌سازی (10 نمره)

نمره اضافه (تا 60 نمره):

- ✓ نمایش گرافیکی درخت هافمن
- ✓ اجرای الگوریتم برای یک تصویر واقعی که در پوشه پروژه قرار داده شده است.
- ✓ پیاده‌سازی حالت بازگشتی و مقایسه با الگوریتم حریصانه.
- ✓ ذخیره مناسب درخت هافمن