

## Veri Yapıları ve Algoritmalar Projesi

Konu: Run Length Encoding (RLE) yöntemi ile görüntü sıkıştırma

**Problem:** Bir görüntüde aynı renkler ardışık olarak çok fazla kullanılıyorsa, rengi gösteren kodu sürekli tekrar etmek yerine kaç defa kullanıldığı ve renk kodu saklanarak yer tasarrufu yapılabilir. Örneğin 5 rengi ardışık olarak 10 defa geçiyorsa 5 5 5 5 5 5 5 5 5 5 yerine bu bilgiyi 10 5 şeklinde saklamak daha avantajlıdır. Burada rengin kaç defa kullanıldığını gösteren 10 sayısı **run length** olarak isimlendirilir.

Bu ödevde **pgm** formatında verilen, renklerin 0-255 arası sayılarla gösterildiği gri bir görüntüyü RLE yöntemi ile sıkıştırarak sıkıştırılmış görüntüyü elde eden ve istendiğinde sıkıştırılmış görüntüyü açarak orijinal görüntüyü elde eden algoritmayı tasarlamamız istenmektedir.

**İşlem Adımları:** Ödev 6 ana bölümden oluşmaktadır:

1. **Görüntü okuma:** pgm dosyanın formatını çözerek görüntünün satır sayısını, sütun sayısını ve görüntüyü okuyunuz. Pgm formatının özellikleri ile ilgili bilgi için aşağıdaki linkten faydalanabilirsiniz.  
<http://netpbm.sourceforge.net/doc/pgm.html>
2. **RLE yöntemi ile görüntü sıkıştırma:** Görüntüye RLE yöntemi uygulayarak bir dosyaya görüntünün satır sayısı, sütun sayısı ve ardışık kullanım adedi ve renk kodu sırası ile sıkıştırılmış bilgiyi bir dosyaya yazınız.
3. **RLE yöntemi ile sıkıştırılmış görüntüyü açma:** Sıkıştırılmış görüntüyü açarak orijinal görüntüyü elde ediniz. Bu işlemi yaparken **önce** sıkıştırılmış görüntünün uygun olup olmadığının kontrolü yapılmalıdır. Bunun için 4. maddede verilen işlemlerin yapılması gereklidir. Eğer bu kontroller başarılı olursa sıkıştırılmış görüntü açılmalıdır.
4. **Kontrol:** Sıkıştırılmış görüntüye ait dosya giriş bilgisi verildiğinde, bunun açılabilir bir sıkıştırılmış veri olup olmadığı aşağıdaki kontroller ile yapılmalıdır:
  - a. Sıkıştırılmış görüntüdeki piksel sayısının  $\text{SatırSayısı} \times \text{SütunSayısı}$  kadar olduğunun kontrolü: Örneğin aşağıdaki örnek sıkıştırılmış görüntü 3 satır ve 2 sütundan yani 6 pikselden oluşmaktadır. 5 adet 7 rengi ve 1 adet 9 rengi kullanıldığı için toplam piksel sayısı 6 olduğundan ötürü doğru bir sıkıştırılmış görüntüdür.  
**Örnek: 3 2 5 7 1 9 (Okunmasının kolay olması için örnekteki sayılar arasında boşluk bırakılmıştır. Dosyayı binary oluşturup sayıları blok olarak okuyup yazmanız daha efektif olacaktır).**
  - b. Sıkıştırılmış görüntüde 0-255 arasındaki renklerin kullanıldığının kontrolü
  - c. Aynı renge ait Run Length'in **arka arkaya** bulunmadığının kontrolü: Örneğin aşağıdaki sıkıştırılmış görüntü 24 pikselden oluşmaktadır. 5 rengi için arka arkaya 2 farklı işlem yapılmıştır. Bu yanlış bir sıkıştırma değildir. Bu şekilde bir dosya gelirse hata mesajı verilmeli ve açma işlemi yapılmamalıdır. Sondaki 3 5 arada başka bir renk olduğu için doğrudur.  
**Örnek: 3 8 6 5 7 5 8 1 3 5**

5. **Görüntü yazma:** Sıkıştırılmış görüntüyü açtıktan sonra pgm formatına uygun halde yeni bir dosyaya yazınız. Bu dosya pgm dosya açabilen herhangi bir görüntü görme aracı ile açılıp görülebilir olmalıdır.
6. **Sıkıştırılmış görüntü üzerinde yapılacak işlemler: Sıkıştırılmış görüntüyü açmadan aşağıdaki işlemleri yapınız:**
- a. **Verilen bir rengin değerini değiştirme:** Örneğin sıkıştırılmış dosyadaki bütün 5 renklerini 10 rengi yapmak gibi.
  - b. **Koordinatları giriş olarak verilen bir pikselin değerini değiştirme:** Örneğin 2. satır 3.sütundaki pikselin değeri değiştirilebilmelidir. Burada sıkıştırılmış görüntünün tamamını açmak yerine, sadece ilgili bölgesinde değişiklik yapmak yeterlidir. Aşağıdaki örnekte resim 5 satır ve 3 sütun olmak üzere toplam 15 pikselden oluşmaktadır. Örneğin (3,1) adresindeki pikselin değeri 90 yapılmak istensin. Her satırda 3 sütun olduğu için (3,1) adresindeki eleman 11. sırada olur. Bu durumda 3 tane 20 değil, 1 90 2 20 olarak değişiklik yapılması gerekir.  
**Örnek:** İşlem Öncesi: 5 3 10 8 3 20 2 6 İşlem Sonrası: 5 3 10 8 1 90 2 20 2 6
  - c. **Resmin histogramının hesaplanması:** Resimdeki her rengin kaç defa kullanıldığı hesaplanmalıdır. Aşağıdaki örnekte resim 4\*4, toplam 16 pikselden oluşmaktadır. Resimde 10 rengi 3, 120 rengi 5, 200 rengi 8 defa geçmiştir.  
**Örnek:** 4 4 3 120 8 200 3 10 2 120