

2020 - 2021 GÜZ Bilgisayar Programlama III ve Bilgisayar Programlama III LAB – Ödev 2

(Seyrek matris sıkıştırma algoritmaları)

Amaç: C’de dizi ve işaretçi (int: pointer) işlemlerini kullanmak ve kütüphane oluşturmak

Seyrek (ing: sparse, sıfır elemanı bol olan) matrislerin sıkıştırılmasında bir takım algoritmalar kullanılmaktadır. Bu ödevde üç adet sıkıştırma algoritmasına değinilecektir. Bunlar sırasıyla Sıkıştırılmış Satır Saklama (ing: Compressed Row Storage, CRS), Sıkıştırılmış Sütun Saklama Sıkıştırılmış Sütun Saklama (ing: Compressed Column Storage, CCS) ve *ij*-Koordinat Sıkıştırması (*ij*-Koordinat Compression) algoritmalarıdır. Ele alınan matris

$$A = \begin{bmatrix} 10 & 0 & 0 & 0 & -2 & 0 \\ 3 & 9 & 0 & 0 & 0 & 3 \\ 0 & 7 & 8 & 7 & 0 & 0 \\ 3 & 0 & 8 & 7 & 5 & 0 \\ 0 & 8 & 0 & 9 & 9 & 13 \\ 0 & 4 & 0 & 0 & 2 & -1 \end{bmatrix}$$

olsun. Bu matrisin sıfırdan farklı elemanlarının tutulduğu tek boyutlu 19 elemanlı dizi

$$nz = [10 \quad -2 \quad 3 \quad 9 \quad 3 \quad \dots \quad 9 \quad 13 \quad 4 \quad 2 \quad -1]$$

biçiminde olur.

1. Sıkıştırılmış Satır Saklama Algoritması:

Bu algoritmada yukarıdaki *nz* ile birlikte iki tane daha tek boyutlu dizi oluşturulur. Bunlardan ilki *nz*’deki elemanların sütun indislerinden oluşup *nz* ile aynı sayıda elemana sahip olan *col_ind* dizisidir. İkincisi ise *A* matrisinin her bir satırının sıfırdan farklı ilk elemanının *nz*’deki indisini tutan *row_ptr* tek boyutlu dizisi olup bu dizinin eleman sayısı *A* matrisinin satır sayısının 1 fazlasıdır. Bu tanımlara göre yukarıda verilen *A* matrisi için ilgili tek boyutlu diziler

$$col_ind = [0 \quad 4 \quad 0 \quad 1 \quad 5 \quad \dots \quad 4 \quad 5 \quad 1 \quad 4 \quad 5]$$

$$row_ptr = [0 \quad 2 \quad 5 \quad 8 \quad 12 \quad 16 \quad 19]$$

olarak elde edilirler. Burada *row_ptr* dizisinin son elemanı *nz* dizisinin son indisi olan 19’dur.

2. Sıkıştırılmış Sütun Saklama Algoritması:

Bu algoritmada yukarıdaki *nz* ile birlikte iki tane daha tek boyutlu dizi oluşturulur. Bunlardan ilki *nz*’deki elemanların satır indislerinden oluşup *nz* ile aynı sayıda elemana sahip olan *row_ind* dizisidir. İkincisi ise *A* matrisinin her bir sütununun sıfırdan farklı ilk elemanının *nz*’deki indisini tutan *col_ptr* tek boyutlu dizisi olup bu dizinin eleman sayısı *A* matrisinin satır sayısının 1 fazlasıdır. Bu tanımlara göre yukarıda verilen *A* matrisi için ilgili tek boyutlu diziler

$$row_ind = [0 \quad 1 \quad 3 \quad 1 \quad 2 \quad \dots \quad 4 \quad 5 \quad 1 \quad 4 \quad 5]$$

$$col_ptr = [0 \quad 3 \quad 7 \quad 9 \quad 12 \quad 16 \quad 19]$$

olarak elde edilirler. Burada *col_ptr* dizisinin son elemanı *nz* dizisinin son indisi olan 19'dur.

3. ij-Koordinat Sıkıştırma Algoritması:

Bu algoritma yukarıdaki iki algoritmaya göre daha basit olup ayrıntıları bu dökümanda verilmeyecektir. İlgili algoritmanın araştırılması ve anlaşılması öğrenciye bırakılmıştır.

Talimatlar:

1. Bu ödevde üç adet dosya ile birlikte compress adında bir kütüphane oluşturmanız istenmektedir. İlgili kütüphane için
 - a. *compress.h* (header file)
 - b. *compress.c* (source file)dosyalarını ve uygulama için de
 - c. *main.c* (main file)adlı dosyaları oluşturmanız beklenmektedir.

2. *compress.h* dosyası içerisinde 6 adet fonksiyon prototipi bulunacaktır. Bunlar:

```
void compress_IJ(int **A, int *nz, int *rows, int *cols)

int **decompress_IJ(int *nz, int *rows, int *cols)

void compress_CRS(int **A, int *nz, int *col_ind, int *row_ptr)

int **decompress_CRS(int *nz, int *col_ind, int *row_ptr)

void compress_CCS(int **A, int *nz, int *row_ind, int *col_ptr)

int **decompress_CCS(int *nz, int *row_ind, int *col_ptr)
```

3. Fonksiyonların adından da anlaşılacağı gibi her biri ilgili sıkıştırma algoritmasını kullanarak sıkıştırma (ing: compress) ve açma (ing: decompress) işlemlerini kullanacaktır.
4. Sıkıştırılmak istenen matrisi *A* ile simgeleneyecektir. *A* matrisinin elemanları 1 ile 10 arasındaki rastgele tamsayılardan oluşacaktır. *A* matrisi oluşturulurken elemanlarının çoğunluğunun 0 olarak alınması önemlidir.
5. Kullanılacak matrisin satır ve sütun sayıları `#define` yardımı ile tanımlanmalıdır.
6. Kütüphaneyi ve ana dosyayı birlikte derleyen *make* dosyası mutlaka oluşturulmalıdır. *make* dosyası ve kütüphane oluşturulmayan ödevler değerlendirilmeyecektir.
7. Ödevler çalışan kod ve rapor olacak şekilde zip'lenerek moodle'da ilgili yere yüklenecektir. Raporun mutlaka bir kapak sayfası olmalı ve bu kapak sayfası öğrencinin adı, soyadı, numarası, dersin adı ve ödevin adını içermelidir.
8. Rapor pdf biçiminde olmalıdır.
9. Rapor kurallarına uymayan raporlar eksikliklerine göre ya değerlendirilmez ya da notları düşürülür.
10. Raporun çıktı olarak verilmesine gerek yoktur.
11. Ödevin teslim tarihi **22 Kasım 2020 Pazar saat 23:55**'tir. Geç gelen ya da mail aracılığıyla gönderilen ödevler kesinlikle kabul edilmeyecektir.