**Pascal Üçgende istenilen satırdaki elemanın bulan algoritmanın akış diyagramı.**

Buradaki yapacağımız işlem sadece üçgendeki iteceğimiz satırdaki bulanan bulunan elemanları bulmaktır.

Pascal'ın üçgeni, üçgenin üst elemanları(sayıların) 1 olduğu ve diğer her sayıların sol üst ve sağ üst tarafındaki sayıların toplamı olduğu, üçgen şeklinde şekillendirilmiş sayılardan oluşan bir kalıptır. Genellikle sol tarafın solundaki boş öğelerin ve 1'lerin sağ tarafının sağının 0 olduğu varsayıldığından bahseder. Sadece resmine bakmak daha kolaydır:

1. satir 1

2. satir 1 1

3. satir 1 2 1

4. satir 1 3 3 1

5. satir 1 4 6 4 1

6. satir 1 5 10 10 5 1

7.satir 1 6 15 20 15 6 1

n. satir \* \* \* \* \* \* \* \* \* \* \* \* \*

Temel olarak, üçgende herhangi bir değerin ne olduğunu bulmak istiyorsanız, öğeleri sol üst ve sağ üstte toplayabilirsiniz. Dolayısıyla, 5. satırdaki 3. öğeyi istiyorsanız, 4. satırdaki 2. ve 3. öğeleri toplayabilirsiniz, bu 3 + 3 = 6 olacaktır. Ayrıca her satırın ilk ve son öğelerinin 1 olacağını da biliyoruz.

Ayrıca, üçgenin değerlerini saklamanın bir yoluna ihtiyacımız var. Kullandığım yol sadece 2 boyutlu bir diziydi. "Ana dizi", üçgenin her bir satırına, her "alt dizi" ise her satırdaki öğelere karşılık gelir. Üçgenin her satırının, satır numarasıyla aynı sayıda öğeye sahip olduğunu not edebiliriz. İlk satırın 1 öğesi [1], ikinci satırın 2. öğesi [1 1], üçüncü satırın 3. [1 2 1] öğesi vardır vb. Dizinin n'inci satırını bulmak istiyorsak, "ana dizi" n adet alt dizi tutması gerekiyor. Ve her alt dizi, satır numarasıyla aynı uzunluğa sahip olacaktır. Dolayısıyla, alt\_dizi\_1 uzunluk 1, alt\_dizi\_2 uzunluk 2, vb.

Artık verilerimizin nasıl yapılandırılacağını bildiğimize göre, geriye kalan tek şey bazı kodları uygulamak ve çalışmasını sağlamaktır. İşte C'de yazdıklarım **/src**/ klasör bulunmaktadır ve /**bin/** klasör içinde compilable hale (makine dilinde) bir dosya bulunmaktadır**. /doc/** klasör içinde Word dosyası, folders tree ve diyagram akışı bulunmaktadır. Bu kod benim githubum üzerinde daha detaylı şekilde bulunmaktadır: ***https://github.com/adjeviarsene/Sayisal-Analiz-odev\_hft02.git.***

Böylece burada kullanıcı yazdırmak istediği satırı girer (number içinde saklanır). myList, her biri setnumber öğelerine sahip olan setnumber alt dizilerine sahiptir. [Not: Bu biraz savurgan ve istenen satır sayısı arttıkça daha israf olur. İdeal olarak, her bir alt dizi, yalnızca gereken minimum boyut olacaktır.

İkili dosyamızı derlemek için terminalde bu komut yazabiliriz:

***gcc -g -fstack-protector-strong -std=c99 pascal.c main.c -o sayısalodevone.out***

Bu durumda gcc kullandım (C compiler), -g flags: bizim ikili dosyamız içinde debugger olarak GDB (Gunu projet Command line debugger) kullanacağımız belirtirdik, -fstack-protector-strong flags kodumuz arka planda çalışırken belleye zarar vermemek için.

Son olarak, önceki bahsettiğimiz tüm aşmaların akış diyagramı bu şekilde bulunmaktadır:

