

Lab 1.1

Getting Started with Julia for cs381

แบบฝึกหัด

- วันนี้เราจะมาลองเขียนโปรแกรม Julia เพื่อประมาณค่า pi
- โดยใช้สมการดังต่อไปนี้ ซึ่งคิดค้นโดยนักคณิตศาสตร์ชาวอินเดีย
- Srinivasa Ramanujan

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

สิ่งที่ต้องทำ

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

- ต้องคำนวณพจน์ที่เป็นอนุกรม
- ต้องหาค่า factorial (ใน Julia มีฟังก์ชันนี้ให้)

การคำนวณอนุกรม

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

- เป็นลักษณะทำซ้ำที่มีจำนวนรอบ (จำกัด หรือ ไม่จำกัด ?)
- นึกถึง for loop

การคำนวณอนุกรม

- ในทางปฏิบัติเราไม่สามารถคำนวณอนุกรมถึงพจน์อนันต์ได้
- เราอาจกำหนดค่า k เป็นค่าที่เหมาะสม ไม่มากหรือน้อยเกินไป เช่น 3

```
for k in 1:3
```

```
    sum += factorial(4*k) * .....
```

```
end
```

ต่อไป

- เมื่อได้ค่าของอนุกรมแล้ว
- เราจะนำมาคูณกับค่าคงที่ด้านหน้า จะได้ ส่วนกลับของค่า pi

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

- แล้วค่า pi มีค่าเท่าไร ?

จะคำนวณอนุกรมถึงพจน์ที่อะไรดี

- 3 ?
 - 5 ?
 - 10 ?
 - 100 ?
 - เราจะตอบคำถามนี้ได้ เราต้องลองเทียบค่า π ซึ่งเป็นที่ยอมรับและถูกนิยามไว้ใน Julia
 - ค่าดังกล่าว เรียกใช้ได้โดยพิมพ์ π
-
- (HINT) เราสามารถพิมพ์สัญลักษณ์กรีกที่ใช้ในคณิตศาสตร์บ่อยๆ ได้โดยพิมพ์ \backslash (backslash) ตามด้วยชื่อสัญลักษณ์แล้วกด tab เช่น $\backslash\pi$ คือการพิมพ์อักษรกรีก π

คำนวณค่าประมาณ pi สำหรับพจน์ที่ 3,5,10,100

```
for k in 1:3
```

```
    sum += factorial(4*k) * .....
```

```
end
```

```
for k in 1:5
```

```
    sum += factorial(4*k) * .....
```

```
end
```

```
.....
```


ใช้ฟังก์ชันดีกว่า

- นิยามฟังก์ชันเพื่อคำนวณค่า pi ที่พจน์ต่าง ๆ กัน

```
function approx_pi(n)
```

```
    for k in 1:n
```

```
        sum += factorial(4*k) ....
```

```
    end
```

```
    return 1/sum
```

```
end
```

คราวนี้คำนวณแล้วเก็บผลลัพธ์ไว้ใน array

```
result = zeros(4) # ประกาศอะเรย์เพื่อเก็บค่า
```

```
for k in 1:10
```

```
    result[k] = approx_pi(k) # คำนวณค่าประมาณ pi โดยที่ k มีค่าต่าง ๆ กัน
```

```
end
```

NOTE: Index ของ Julia เริ่มที่ 1 ไม่ใช่ 0

Plotting the diff

- เราสามารถนำค่าผลต่างที่ได้มาพล็อตกราฟโดยใช้ Winston library
- Try it out.