

Алгоритмыг шинжлэх, үнэлэх

Б.Наранчимэг

Мэдээлэл, компьютерийн ухааны тэнхим

ХШУИС, МУИС

naranchimeg@seas.num.edu.mn

Өмнөх хичээлээр

- Алгоритмлах үе шат
 - Алгоритмыг зохиох
 - **Алгоритмыг шалгах**
 - Алгоритмыг шинжлэх

Алгоритм шалгах

- Validation vs. Verification
- Counter example
- Тестийн тохиолдлуудыг тодорхойлох (test case)
- Тестийн өгөгдлийг тодорхойлох (test data)
- Хар хайрцагны тест (Black box testing)
 - Тэнцүү хуваах ба захын утгын шинжилгээний арга
- Цагаан хайрцагны тест (White box testing)
 - Удирдлагын урсгалын тест (Control-flow testing)
 - Өгөгдлийн урсгалын тест (Data-flow testing)

Практикт хэрэглэгдэх аргууд

- Утгын хүснэгтийн арга
- Логик шалгалтын арга
- Хэсэгчилэн шалгах арга

Агуулга

- Алгоритмлах үе шат
 - Алгоритмыг зохиох
 - Алгоритмыг шалгах
 - Алгоритмыг шинжлэх

Алгоритмыг шинжлэх

- Алгоритмын хэмжигдэхүүний нийт тоо буюу санах ойн зарцуулалт (memory)
- Алгоритмын биелэгдэх хугацаа (computational time)
- Алгоритмыг хэрэгжүүлэх үед гарах зардлын тооцоо

Алгоритмыг шинжлэх

- Алгоритмын шинжилгээ яагаад хэрэгтэй вэ?
 - Зөвхөн ажилладаг, зөв програм бичих нь хангалтгүй
 - Хэрвээ програм их хэмжээний өгөгдөлтэй ажилдаг бол ажиллах хугацаа нь асуудал болж хувирна.

Жишээ

- Өгсөн N ширхэг тооны жагсаалтаас K дахь хамгийн их элементийг ол.
- **Алгоритм 1**
 1. N тоог массивт уншиж авах
 2. Зарим энгийн алгоритмыг ашиглан массивын элементийг буурах дарааллаар эрэмбэлэх
 3. K байрлал дахь элементийг үр дүн болгон буцаах

Жишээ

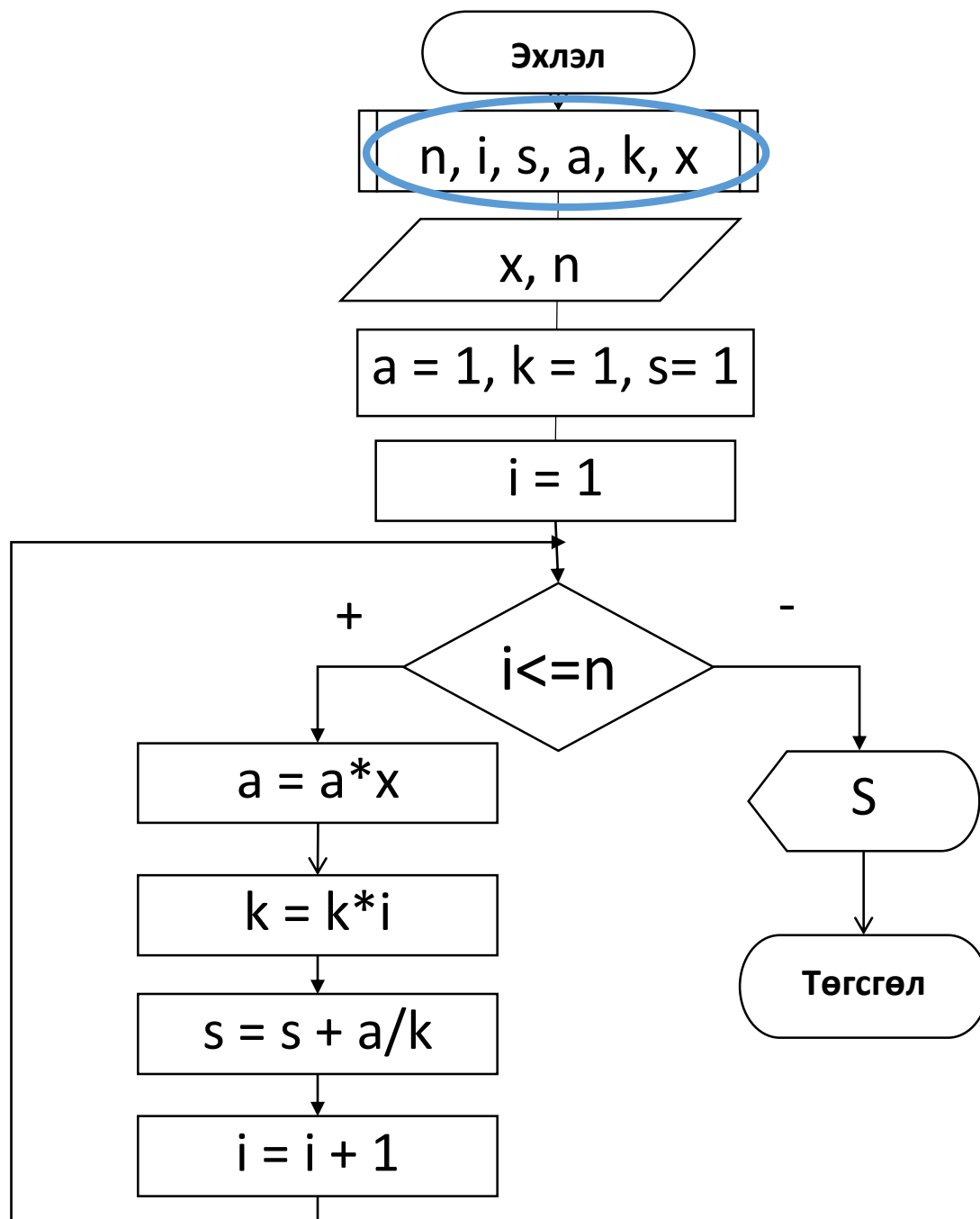
- Өгсөн N ширхэг тооны жагсаалтаас K дахь хамгийн их элементийг ол.
- Алгоритм 2
 1. Эхний k ширхэг элементийг массивт оруулж буурах дарааллаар эрэмбэлэх
 2. Үлдсэн элементийг нэг нэгээр нь уншиж
 1. Хэрвээ K дэхь элементээс бага бол алгасах
 2. Үгүй бол массивын зөв талбарт байгаа элементийг оруулах
 3. K байрлал дахь элементийг үр дүн болгон буцаах

Жишээ

- Аль алгоритм нь дээр вэ?
 - $N = 100$ ба $k = 100$?
 - $N = 100$ ба $k = 1$?
- $N = 1,000,000$ ба $k = 500,000$ байхад юу болох вэ?
- Илүү сайн алгоритмууд байдаг

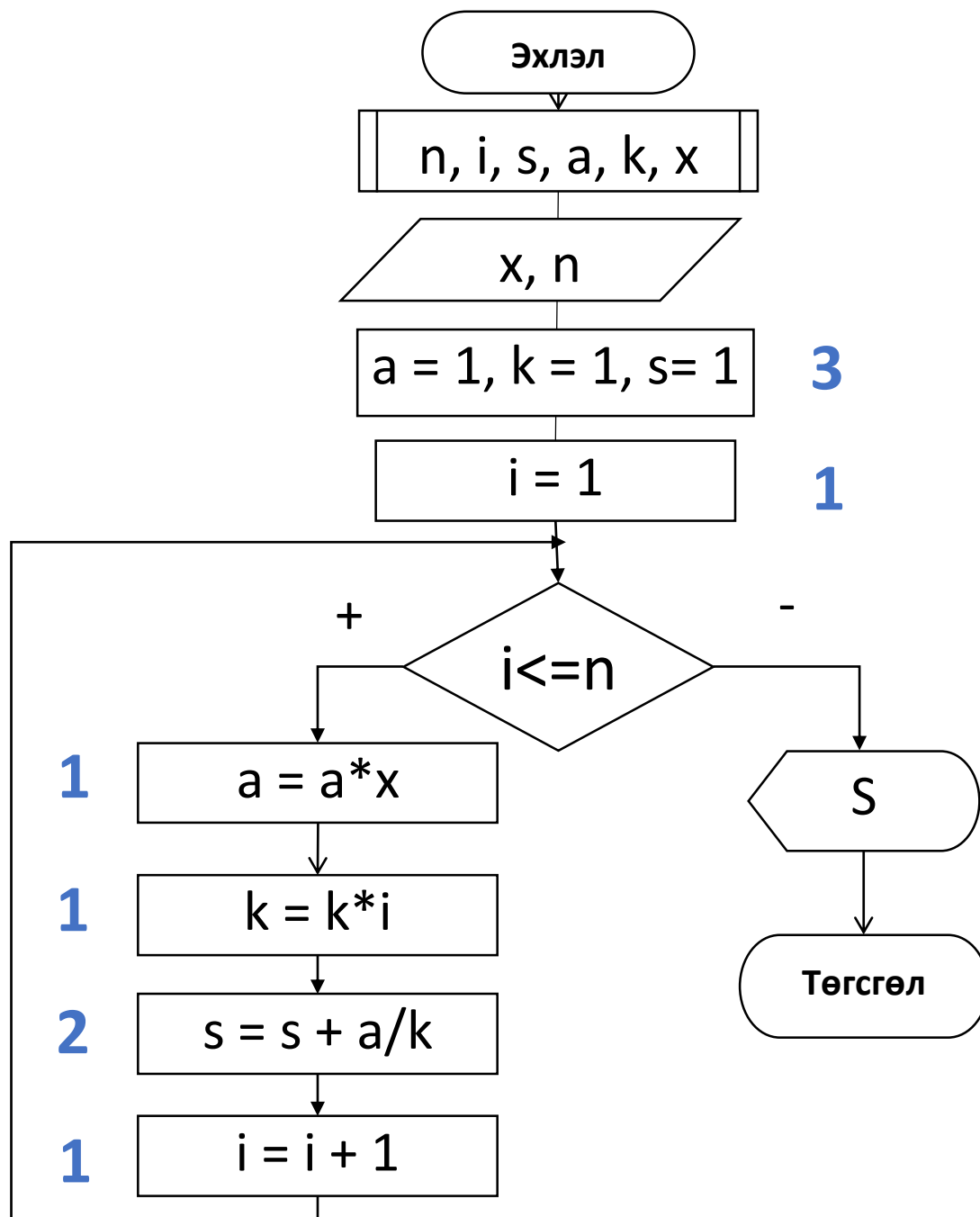
Алгоритмыг шинжлэх

- Энэ шийдэл хэр сайн вэ?
 - "Миний програм 1.37 секундын дотор 2-оос 1,000,000,000 хүртэлх бүх анхны тоог олдог."

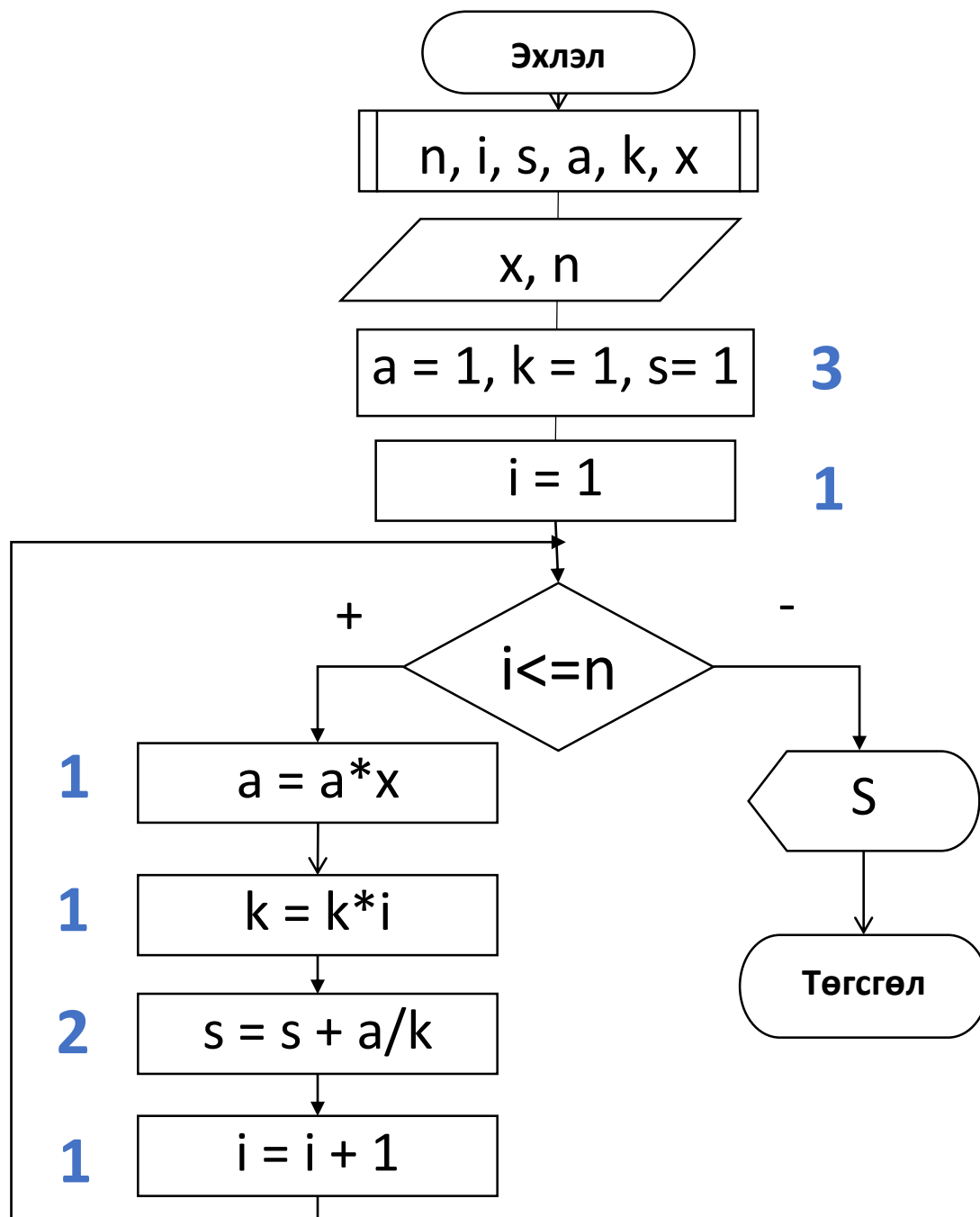


$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$

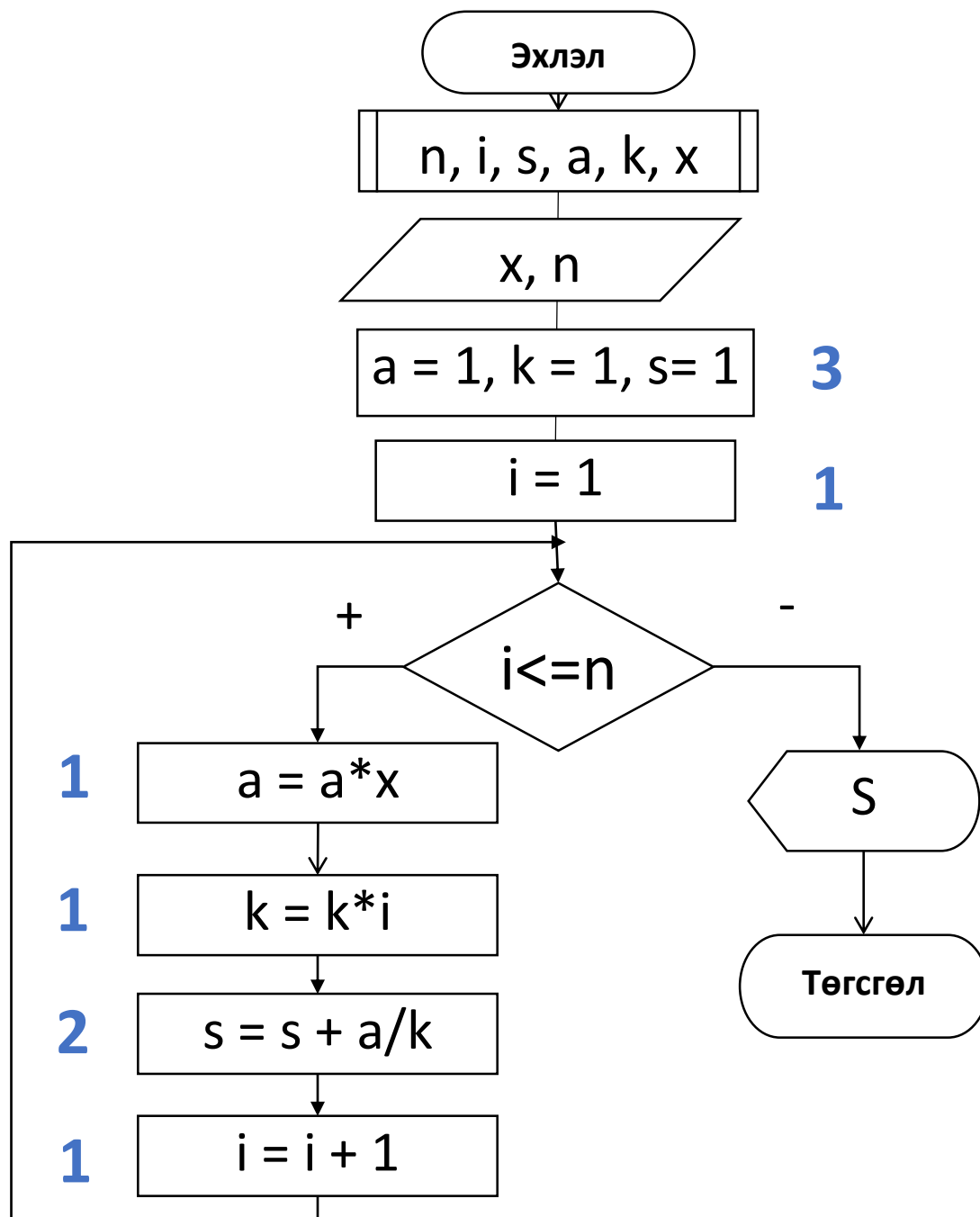
6 хэмжигдэхүүн байна.



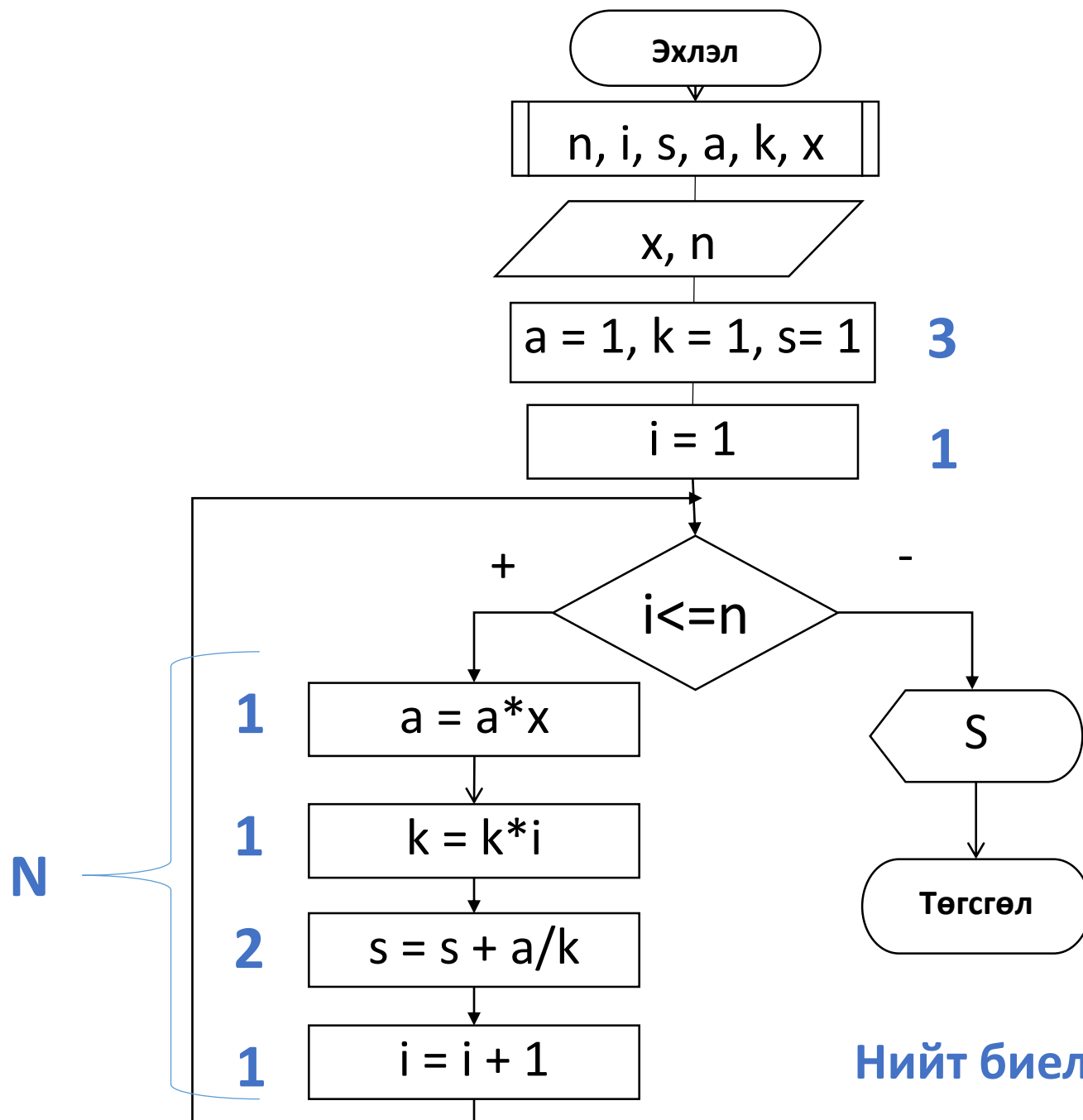
$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$



$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$



$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$



$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$

Нийт биелэгдэх хугацаа: $5N+4$

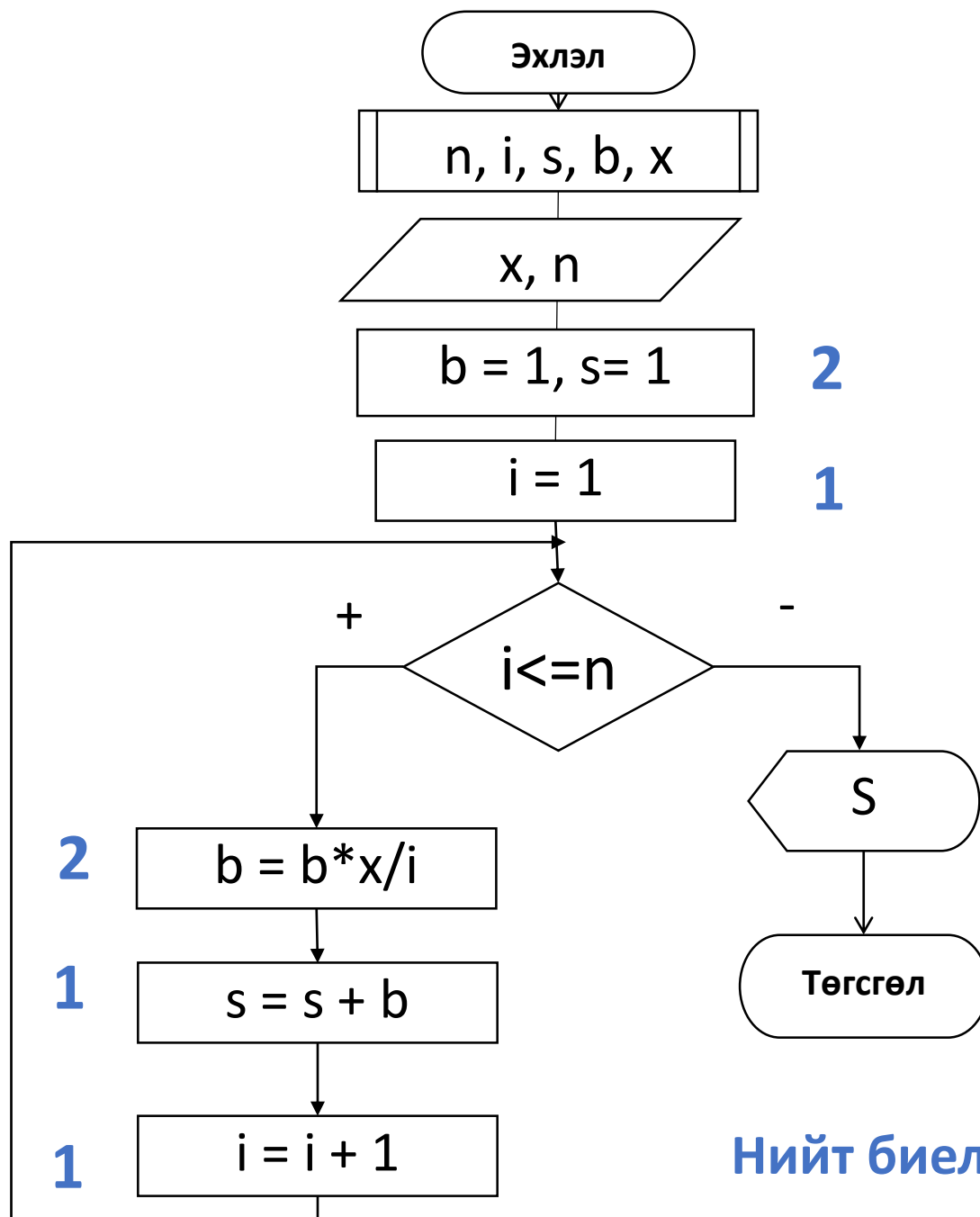
Сайжруулах нь

- Энэхүү шинжилгээг хийсний үр дүнд алгоритмыг сайжруулах буюу илүү сайн алгоритм зохиож болно.
- Рекурент томъёо

$$\frac{x^i}{i!} = \frac{x^{i-1}}{(i-1)!} \cdot \frac{x}{i}$$



$$b = b \cdot \frac{x}{i}$$

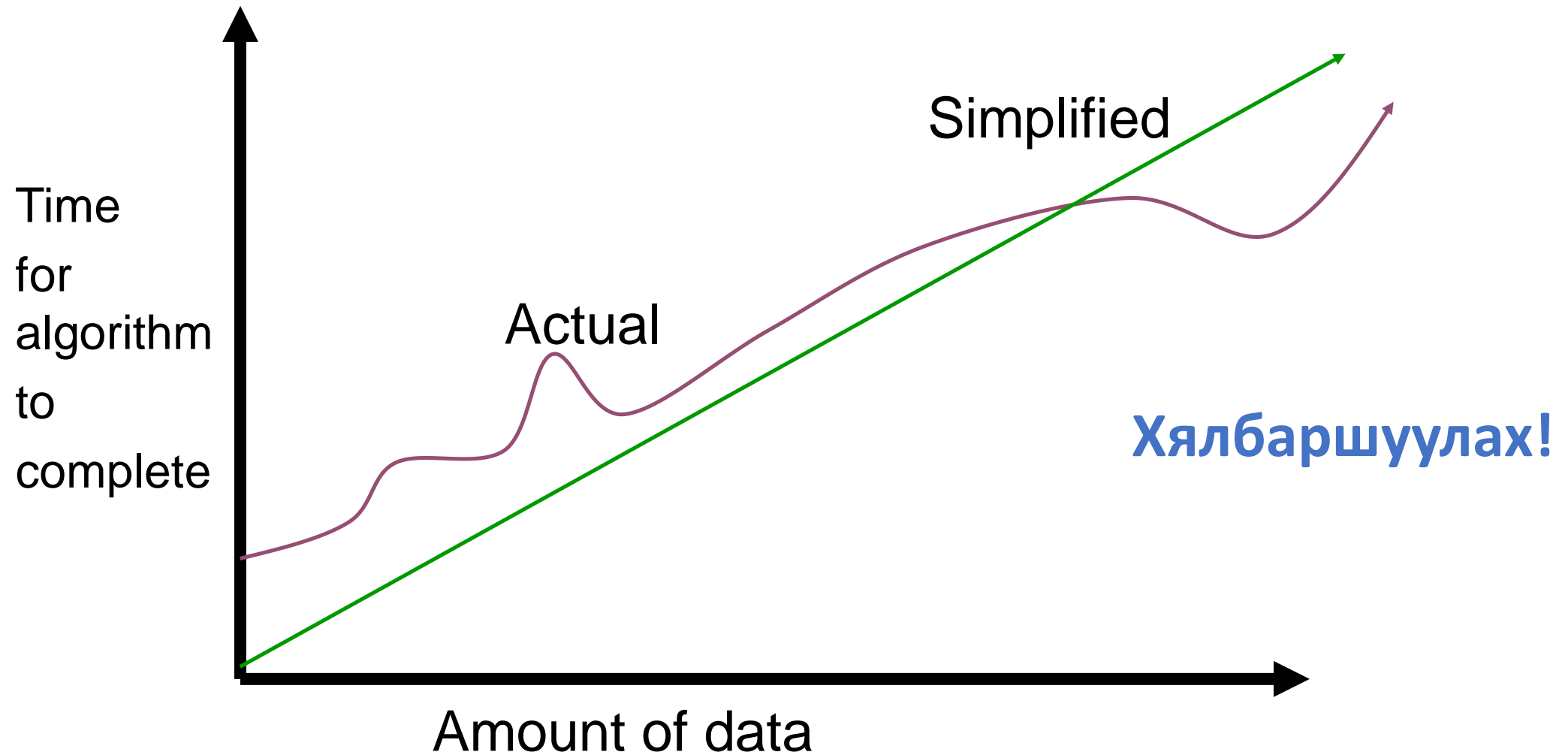


$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$

5 хэмжигдэхүүн байна.

Нийт биелэгдэх хугацаа: $4N+3$

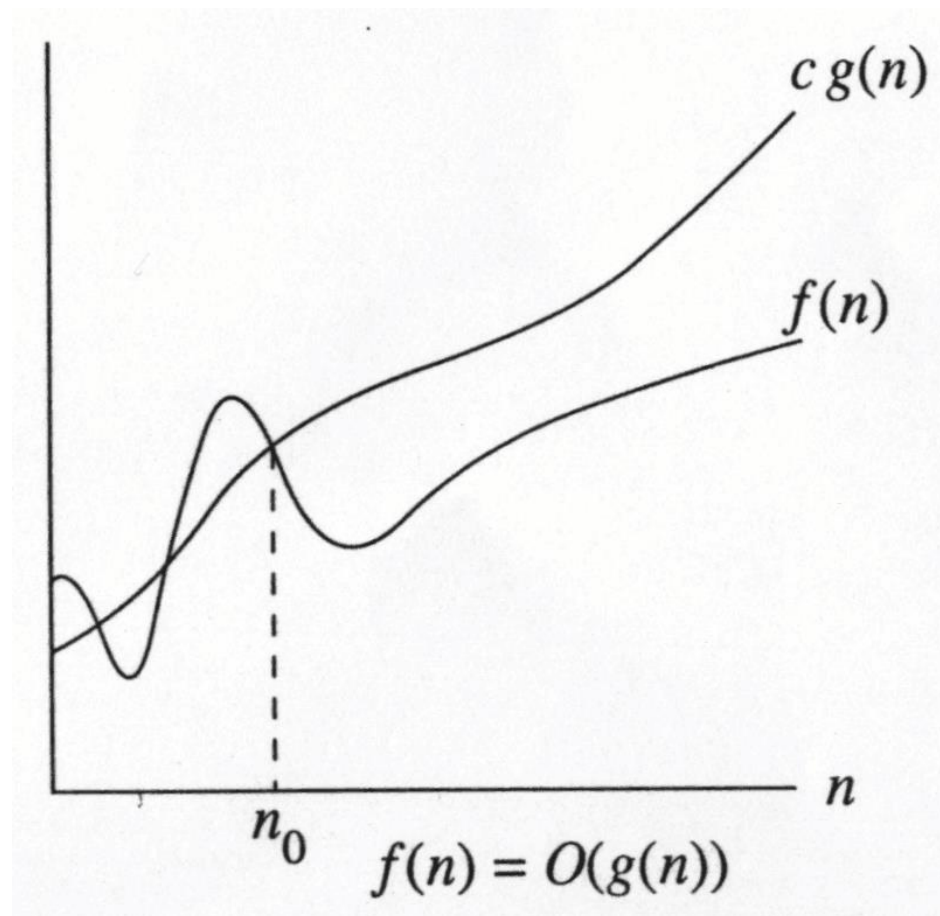
Actual vs. Simplified



Big O буюу Том Өү

- Алгоритмын гүйцэтгэх хугацааг илэрхийлдэг хамгийн түгээмэл арга
- $f(N) = O(g(N))$
- $f(N) \leq c g(N)$ when $N \geq n_0$
- c болон n_0 тогтмолууд олдож байвал
- $F(N)$ -ийн өсөлтийн хурд $g(N)$ -ээс бага буюу тэнцүү
- $g(N)$ нь $f(N)$ -ийн дээд хязгаар гэж үзнэ

Өсөлтийн хурд



Жишээ

- Хэрвээ $f(N) = 4N + 3$ бол
 - $f(N) = O(N)$
- $f(N) = 2N^2$ бол
 - $f(N) = O(N^2)$
- $f(x) = 6x^4 - 2x^3 + 5$
 - $f(x) = O(x^4)$

Бусад тооцооллын аргууд

- O : Asymptotic Upper Bound
- Ω : Asymptotic Lower Bound
- Θ : Asymptotic Tight Bound

Алгоритмыг шинжлэх

- $T(N)$: алгоритмын бодит өсөлтийн хурд юм
 - програмын эсвэл хэсэгчилсэн кодын гүйцэтгэгдэж болох үйлдлийн тоотой тэнцүү
- $F(N)$: өсөлтийн хурдыг хязгаарлах функц
 - дээд эсвэл доод хязгаар
- $T(N)$ нь заавал $F(N)$ -тэй тэнцүү байж болохгүй.
 - тогтмол ба бага нэр томъёог тооцоолохгүй байж болно.

Бусад тооцооллын аргууд

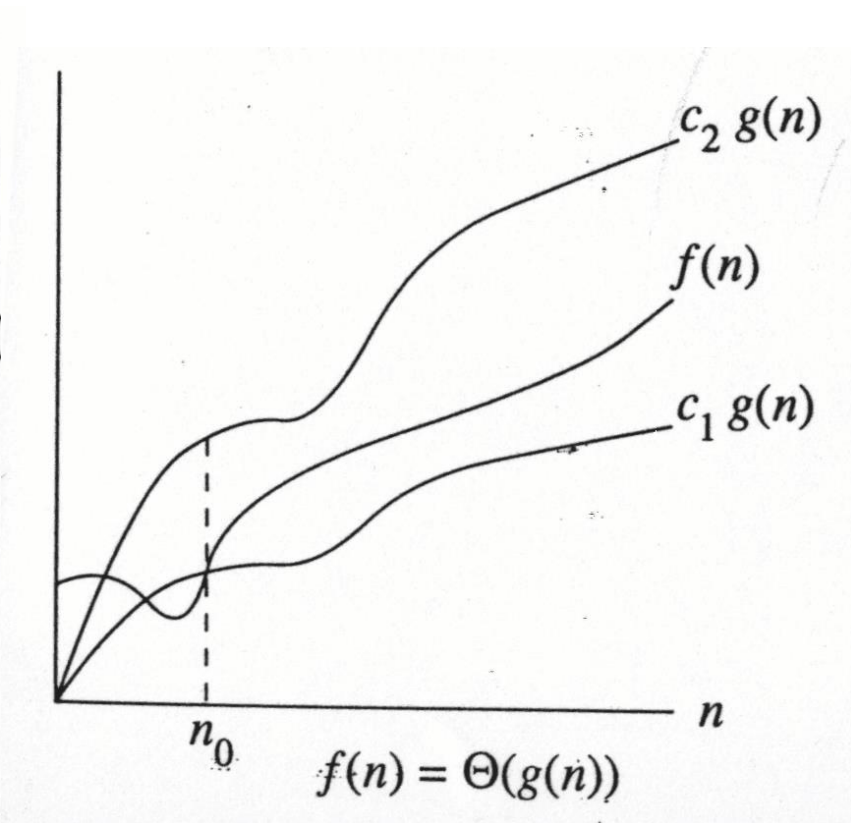
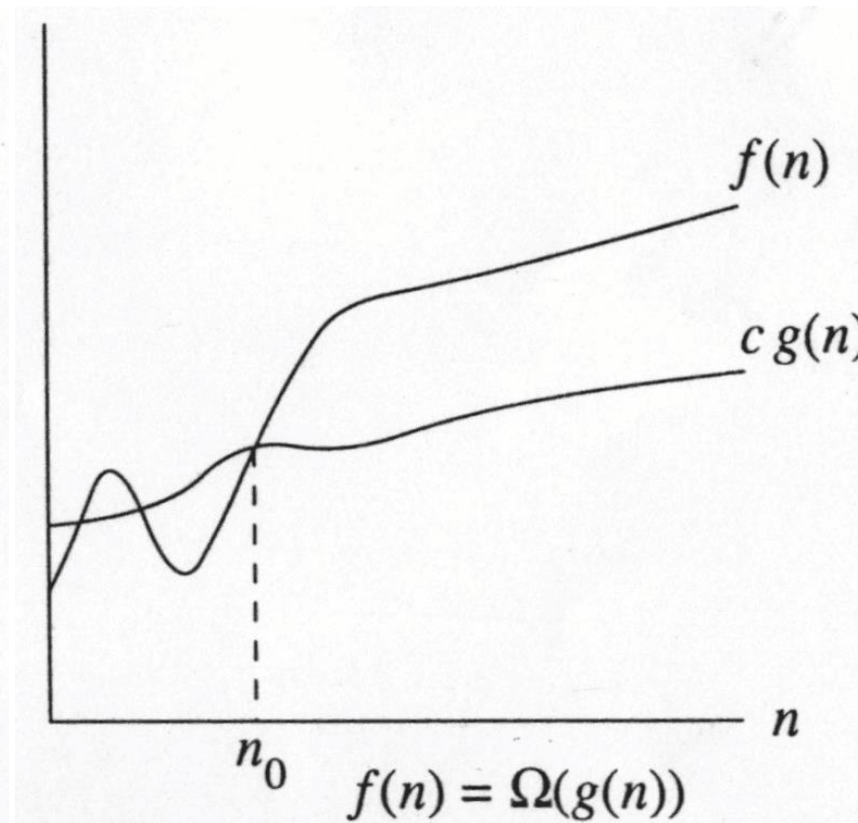
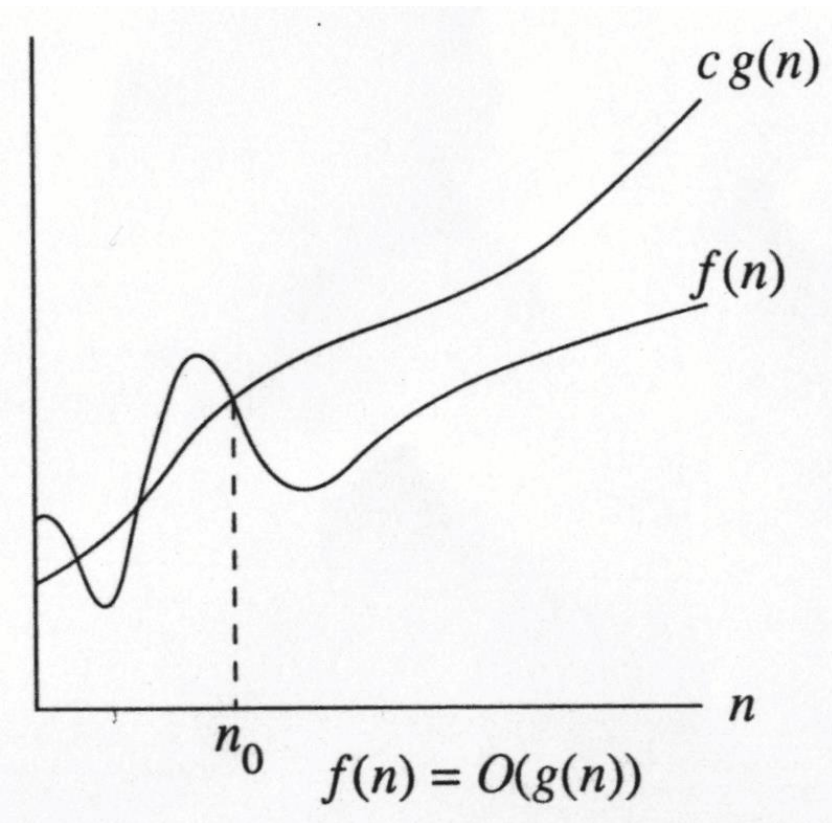
Analysis Type	Mathematical Expression	Relative Rates of Growth
Big O	$T(N) = O(F(N))$	$T(N) \leq F(N)$
Big Ω	$T(N) = \Omega(F(N))$	$T(N) \geq F(N)$
Big θ	$T(N) = \theta(F(N))$	$T(N) = F(N)$

Big O нь дээд хязгаараас бага эсвэл тэнцүү байна

Big Omega нь доод хязгаараас их эсвэл тэнцүү, ижил төстэй байдаг

Big Theta нь тэнцүүтэй төстэй

Big O vs Big omega vs Big theta



Жишээ нь

- $f(n) = 4n^3 + 10n^2 + 5n + 1$
- $g(n) = n^3$
- Big O
 - $f(n) \leq 5 \cdot g(n)$
 - $O(g(n)) = O(n^3)$
- Big Omega
 - $f(n) \geq 4 \cdot g(n)$
 - $\Omega(g(n)) = \Omega(n^3)$
- Big Theta
 - $4 \cdot g(n) \leq f(n) \leq 5 \cdot g(n)$
 $\theta(g(n)) = \theta(n^3)$

Шинжилгээний аргууд

- Шууд тооцоолох арга
 - Алхам, үйлдлүүдийг шууд тооцоолох
- Ойролцоогоор үнэлэх арга
 - Хамгийн хурдан (best case)
 - Хамгийн удаан (worst case)
 - Дундаж биелэгдэх хугацаа (average case)