Mathematics of Computer Science. - M.I.T. opencourseware

Abstract:
In this lecture we learned the rest part of sums and asymptotic notation. I was never taught the nature log so the lecture was really strange to me but soon get used to it. Sums is used for find a value's future amount in rough silhouette. To figuring the amount, we use the integration bounds and stirling formula. Asymptotic notation play a role of how function grows in the limit in fair condition.

For the English:
diverge, asymptotic, formula, equation, polynomial, bizarre, oscillate, monotonically, got off track, nastier, notation, strictly

Lec 13. SUMS AND ASYMPTOTICS:
● **Tons of equations so I cut off most of them and just write down the big picture of method**

- Greedy strategy
Given n-blocks of length 1, we stack up blocks on edge of table so we make the block off from the edge of table in 1 length
Stability constraint : the center of mass Ck of the top K-blocks must lie on the (K+1)st block
for greedy stacking, Ck = rk+1

the center of mass of the Kth block is at rk-1/2
the center of mass of the top k blocks is Ck = {(k-1)Ck-1 + 1(rk − 1/2)}/k-1+1 = {(k-1)Ck-1 + rk − 1/2}/k
rk+1 = rk − 1/2k => rk − rk+1 = 1/2k

perturbation method => r1 − rn+1 = ∑1/2i // rn+1 = 0 : table
r1 = 1/2 * ∑1/i = Harmonic sums
Nth harmonic number => Hn = ∑1/i and then H4 = 25/12 > 2 --- <u>can be off the table</u>
However we could not get a precise formula of Hn so we use Integration bounds to figure amount roughly

f(n) + ∫f(x)dx =< ∑f(i) =< f(1) + ∫f(x)dx
differentiate
f(i) * 1/i // ∫dx/x = ln(x)|n~1 => ln(n)
1/n + ln(n) =< Hn =< 1 + ln(n)
Hn ~ ln(n) => Hn = ln(n) + 1/2n + 1/12n^2 + ∑(n)/120n^4 + d (Euler's constant 0.577..)
where ∀n 0 < ∑(n) < 1

N! = ∏i => ln(n!) = ln(1*2*3...*n) = ln(1) + ln(2) ... ln(n) = ∑ln(i)
Integration bounds for increasing sums f(i) = ln(i)
f(1) + ∫f(x)dx =< ∑f(i) =< f(n) + ∫f(x)dx
(xln(x) - x)|n~1 = nln(n) - n + 1
0 + nln(n) - n + 1 =< ln(n!) =< ln(n) + nln(n) - n + 1
=> n^n/e^n-1 =< n! =< n^n+1/e^n-1 (Bounds)

- Stirling's formula

n! = (n/e)^n * √2πn * e^E(n) - > epsilon   where <u>1/12n+1 =< e(n) =< 1/12n</u>

but the epsilon is meaninglessly small (if n = 100 -> E(100) = 1/1200)

Tilde => n! ~ (n/e)^n * √2πn we could roughly omit the epsilon

- Asymptotic Notation : how function grows in the limit

- Tilde : f(x) ~ g(x) if L f(x)/g(x) = 1
- Big O notation (small o) : f(x) = O(g(x)) if L |f(x)/g(x)| < ∝ (finite and can't diverge)
  - L'hopital rule : any polynomial grows slower than any exponential
  - Time to multiple n*n matrices => O(n^3) - cubic
- Omega symbol (small omega) : f(x) = Ω(g(x)) if L|f(x)/g(x)| > 0
- Theta symbol : f(x) = Θ(g(x)) of L |f(x)/g(x)| < ∝ && L|f(x)/g(x)| > 0 (both hold, little but strictly)