**Problem**: Arithmetic on integers $a, b$, two $n$-digit numbers, for large $n$.

**Note**: Naive addition takes linear time, multiplication quadratic.

Can we speed up multiplication at the expense of more additions?

Let $a = 2,345 = 23 \times 10^2 + 45$

$b = 6,789 = 67 \times 10^2 + 89$

$p_1 = a_1 b_1 = 23 \times 67 = 1,541$

$p_2 = a_2 b_2 = 45 \times 89 = 4,005$

$p_3 = (a_1 + a_2) (b_1 + b_2) = (23 + 45)(67 + 89) = 10,608$

$a \times b = p_1 \times 10^4 + (p_3 - p_2 - p_1) \times 10^2 + p_2 = 15,920,205$

Noting that multiplying by $10^2$ and $10^4$ (really powers of 2) can be accomplished by shifting. So can the following **mod** operation. The above uses 3 multiplications. Divide-&-conquer suggests we use the same idea to do the multiplications in computing the $p_i$.

```plaintext
function mult(a, b : large_int) : large_int;
    n := number of digits in smaller of a, b
    if n sufficiently small then return(a * b)
    s := n div 2
    a1 := a div 10^s
    a2 := a mod 10^s
    b1 := b div 10^s
    b2 := b mod 10^s
    p1 := mult(a1, b1)
    p2 := mult(a2, b2)
    p3 := mult(plus(a1, a2), plus(b1, b2))
    return(p1 * 10^2s + (p3 - p2 - p1) * 10^s + p2)
```

**Analysis**: Let $M(n)$ - time to perform `mult` on two $n$-digit numbers

$$M(n) = 3M\left(\frac{n}{2}\right) + c \times n \quad \text{(recurrence)}$$

How do we get that $M(n) = O(n^{1.58496...})$?

**Recurrences**

We'll do 3 classes of techniques:
- summing factors,
- characteristic roots,
- generating functions.

**Summing Factors**

Easiest recurrence: $M(n) = c \times M(n-1)$ if $n > 0$, $M(0) = 1$

**first-order** because $M(n+1) = f(M(n))$, and not $M(n-1),...,M(1)$
homogeneous because (aside from boundary conditions) \( M(n) = 0 \) for all \( n \) is a solution

linear because linear in \( M(i) \) (no terms like \( M(i)^3 \))

\[ M(1) = c, \quad M(2) = c^2, \quad \ldots, \quad M(n) = c^n \sim \text{solution} \]

Next easiest_: variable multipliers. Given \( b_1, b_2, \ldots \)

\[ M(n+1) = b_{n+1} M(n), \quad M(0) = \ldots, \quad \text{solution} = M(n+1) = M(0) \prod_{1 \leq k \leq n+1} b_k \]

Towers of Hanoi_: Let \( M(n) \) - minimal # moves to transfer \( n \) rings. As an upper bound,

- Move \( n-1 \) rings to surplus pole
- Move biggest ring to destination pole
- Move \( n-1 \) rings to destination pole

Let \( A(n) \) - # moves of above algorithm

\[ A(n) = \begin{cases} &2^*A(n-1) + 1, \quad \text{if } n > 0 \\ &0 \end{cases} \]

Linear first-order nonhomogeneous _ recurrences: Given \( b_1, b_2, \ldots, c_1, c_2, \ldots \)

\[ M(n+1) = b_{n+1} M(n) + c_{n+1}, \quad M(0) = \ldots \]

\[ M(1) = b_1 M(0) + c_1, \quad M(2) = b_2 b_1 M(0) + b_2 c_1 + c_2, \ldots \quad \text{ugh} \]

- Introduce a change of variables

\[ X(n) = \frac{M(n)}{b_n \ldots b_1}, \quad X(0) = M(0) \]

defines a new sequence of unknowns \( X(0), X(1), \ldots \)

- Substitute for \( M(n) = b_n \ldots b_1 X(n) \) & solve

\[ b_{n+1} b_n \ldots b_1 X(n+1) = b_{n+1} b_n \ldots b_1 X(n) + c_{n+1} \]

\[ X(n+1) = X(n) + \frac{c_{n+1}}{b_{n+1} b_n \ldots b_1} \]

\[ X(n) = X(0) + \sum_{1 \leq k \leq n} \frac{c_k}{b_k \ldots b_1} \]

- Substitute back for original unknowns

\[ M(n) = b_n \ldots b_1 \left( M(0) + \sum_{1 \leq k \leq n} \frac{c_k}{b_k \ldots b_1} \right) \]

Ex: Return to \( A(n) \), where \( b_k = 2, \quad c_k = 1, \quad k \geq 1 \).

\[ A(n) = 2^n \sum_{1 \leq k \leq n} \frac{c_k}{b_k \ldots b_1} = 2^n \left( \sum_{1 \leq k \leq n} \left( \frac{1}{2} \right)^k \right) = 2^n \left[ \sum_{0 \leq k \leq n} \left( \frac{1}{2} \right)^k - 1 \right] = 2^n \left[ \frac{1 - (1/2)^{n+1}}{1 - (1/2)} - 1 \right] = 2^{n-1} \]

Ex: Consider worst-case of SelectionSort \( r[1..n] \)

procedure SelectionSort(var r, lo, n);

\{r[1..lo-1] already sorted, \( r[i] \leq r[j] \) for \( 1 \leq i < lo \leq j \leq n \} \]

var i : lo..up;
begin
if lo < n then begin

\[ i := \min(r, lo, n); \quad \{ r[i] \leq r[j], \quad lo \leq j \leq n \} \]
swap(r[lo],r[i]);
SelectionSort(r,lo+1,n)
end

Computing \( \min(x_1,...x_n) \) using pairwise comparisons uses (best case) at least \( n-1 \) comparisons. Why? Assume elements distinct. Every non-\( \min \) (there are \( n-1 \) of them) must "win" (be the larger) a comparison.

**Adversary Argument**: Suppose \( x_i \) chosen, but \( x_j \) never won, \( i \neq j \). Algorithm would perform the same way (it'd choose \( x_i \)) with \( x_j \) replaced by \(-\infty\), but it'd be wrong.

**Analysis** of SelectionSort: Let \( M(n) \) be operations used by SelectionSort and assume \( \min \) takes \( p^*n \) operations for \( n \) elements.

\[
M(n+1)=M(n)+p^*(n+1) \quad \text{for} \quad n > 1, \quad M(0)=1
\]

\[
M(n) = b_n \ldots b_1 \left( M(0) + \sum_{1 \leq k \leq n} \frac{c_k}{b_k \ldots b_1} \right) \quad b_k=1, c_k= p^*k
\]

\[
M(n) = 1 + \sum_{1 \leq k \leq n} p^* k = 1 + \frac{p^* n^* (n+1)}{2}
\]

Linear second-order homogeneous recurrence:

**Ex**: Solve the recurrence \( t_n - 5t_{n-1} + 6t_{n-2}=0 \), \( t_0=0, t_1=7 \)

Being second order, can't use summing factors. Must use

**Characteristic Roots**

Assuming constant coefficients:

\[
t_n + c_1 t_{n-1} + \ldots + c_k t_{n-k} = 0
\]

\[
t_0 = \ldots, t_{k-1}=\ldots
\]

**Theorem**: (Superposition) Given the linearity of the recurrence, if \( t_n=f(n) \) and \( t_n=g(n) \) are solutions, then so is \( t_n=\alpha f(n)+\beta g(n) \) for arbitrary \( \alpha \), \( \beta \).

**Proof**: \[
[\alpha f(n)+ \beta g(n)]+c_1[\alpha f(n-1)+ \beta g(n-1)]+\ldots+c_k[\alpha f(n-k)+ \beta g(n-k)]=
\alpha[f(n)+c_1 f(n-1)+c_k f(n-k)] + \beta [g(n)+c_1 g(n-1)+c_k g(n-k)]
\]

\[
\alpha*0 + \beta \cdot 0 = 0
\]

**Ex**: Solve the recurrence \( t_n - 5t_{n-1} + 6t_{n-2}=0 \), \( t_0=0, t_1=7 \)

- "Guess" the form of the solution. \( t_n = x^n \)

Solve \( x^n - 5x^{n-1} + 6x^{n-2} = 0 \)

Not interested in solutions \( x=0 \) (\( t_n=0 \)), so divide through by \( x^{n-2} \) to obtain the characteristic equation for the recurrence \( x^2 - 5x + 6 = 0 \)

- Find the characteristic roots: \((x-3)(x-2)=0 \Rightarrow x=3, x=2\)

Solution will be of the form \( \alpha 3^n + \beta 2^n \)
- Use initial conditions to solve for \( a \), \( b \).

\[
0 = t_0 = a, \quad 2^0 = a + b
\]
\[
7 = t_1 = a, \quad 2^1 = 3a + 2b \quad \alpha = -b, \quad 7 = -3b \quad +2b = -b \quad a = 7
\]

Solution: \( t_n = 7n^3 - 7n^2 \) and it fits.

In general:

- Write down characteristic equation \( x^k + c_1x^{k-1} + \ldots + c_k = 0 \)
- Find characteristic roots \( x_1, \ldots, x_k \) and form general solution \( t_n = a_1x_1^n + a_2x_2^n + \ldots + a_kx_k^n \)
- Use the \( k \) initial conditions to solve for the \( a_i \), \( 1 \leq i \leq k \)

**Ex:**
Given a server & a queue, during a small time \( dt \):

- \( \Pr\{\text{arrival in } dt\} = \lambda dt \)
- \( \Pr\{>1 \text{ arrival in } dt\} = o(dt) \) i.e., negligible
- \( \Pr\{1 \text{ client served in } dt\} = \mu dt \)
- \( \Pr\{>1 \text{ client served in } dt\} = o(dt) \) i.e., negligible

For \( n \geq 1 \), \( \Pr\{\text{queue of length } n\} \) at time \( t + dt \) is:

\[
(t - \lambda dt - \mu dt) \cdot p_n(t) + \lambda dt \cdot p_{n-1}(t) + \mu dt \cdot p_{n+1}(t)
\]

Steady state solution:

\[
\frac{dp_n(t)}{dt} = 0
\]

\[
\frac{dp_n(t)}{dt} = p_n(t + dt) - p_n(t) = -\lambda p_n(t) + \mu * p_{n-1}(t) + \mu * p_{n+1}(t)
\]

\[
\frac{\mu}{p_{n+1}} - (\lambda + \mu) * p_n + \lambda * p_{n-1} = 0, \quad n \geq 1
\]

Characteristic equation:

\[
x^2 - \left(\frac{\lambda}{\mu}\right)x + \frac{\lambda}{\mu} = 0
\]

Let \( \rho = \frac{\lambda}{\mu} \quad \Rightarrow \quad x^2 - (1 + \rho)x + \rho = 0
\]

Characteristic roots: \( x = 1, \quad x = \rho \quad \Rightarrow \quad p_n = \alpha n + \beta \rho^n \quad n = \alpha + \beta \rho^n \)

Also:

\[
p_0(t + dt) = (1 - \lambda dt) \cdot p_0(t) + \mu \cdot dt \cdot p_1(t)
\]

In steady state:

\[
\frac{dp_0}{dt} = 0
\]

\[
\frac{dp_0}{dt} = p_0(t + dt) - p_0(t) = -\lambda p_0(t) + \mu \cdot p_1(t)
\]

\[
\lambda * p_0 = \mu * p_1 \quad \Rightarrow \quad \lambda * (\alpha + \beta \rho^0) = \mu \cdot (\alpha + \beta \rho^1)
\]

\[
\alpha \cdot (1 - \rho) = 0 \quad \Rightarrow \quad \alpha = 0 \quad \text{(for } \rho \neq 1) \quad \Rightarrow \quad p_n = \beta \rho^n
\]

Need 1 more initial condition:

\[
\sum_{0 \leq n} p_n = 1
\]

\[
\sum_{0 \leq n} \beta \rho^n = 1 \Rightarrow \beta \left(\frac{1}{1 - \rho}\right) = 1 \quad \Rightarrow \quad \beta = 1 - \rho \quad \Rightarrow \quad p_n = (1 - \rho) \rho^n
\]

For \( \rho \geq 1 \), no steady state solution (\( p_n \leq 0, 0 \leq n \))

\[E[\text{queue length}] =\]
\[
\sum_{0 \leq n} n \cdot \Pr(\text{queue length } = n) = \sum_{0 \leq n} n(1 - \rho)\rho^n = (1 - \rho) \sum_{0 \leq n} n\rho^n = \frac{(1 - \rho)\rho}{(1 - \rho)^2} = \frac{\rho}{(1 - \rho)}
\]

What if characteristic roots contain irrational numbers?

**Ex:** \( t_n = t_{n-1} + t_{n-2} \), \( t_0 = 0, t_1 = 1 \) 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

- \( x^2 - x - 1 = 0 \)
  
  - roots  
  \[ x = \frac{1 \pm \sqrt{1 + 4}}{2} = \frac{1 \pm \sqrt{5}}{2} \]
  \[ x_1 = \frac{1 + \sqrt{5}}{2}, \quad x_2 = \frac{1 - \sqrt{5}}{2} \]
  
  - \( 0 = t_0 = \alpha \left( \frac{1 + \sqrt{5}}{2} \right)^0 + \beta \left( \frac{1 - \sqrt{5}}{2} \right)^0 = \alpha + \beta \Rightarrow \alpha = -\beta \)

\[
t_1 = \alpha \left( \frac{1 + \sqrt{5}}{2} \right)^1 + \beta \left( \frac{1 - \sqrt{5}}{2} \right)^1 = -\beta \left( \frac{1 + \sqrt{5}}{2} \right) + \beta \left( \frac{1 - \sqrt{5}}{2} \right) \Rightarrow \alpha = \frac{1}{\sqrt{5}}, \beta = -\frac{1}{\sqrt{5}}
\]

\[
t_n = \frac{1}{\sqrt{5}} \left[ \left( \frac{1 + \sqrt{5}}{2} \right)^n - \left( \frac{1 - \sqrt{5}}{2} \right)^n \right]
\]

Note that \( t_n \) always contains integers. Also typically largest characteristic root will dominate as \( n \to \infty \).

(define (fib n)
  (\( \star \) (/ 1 (sqrt 5)))
  (- (expt (/ (+ 1 (sqrt 5)) 2) n) (expt (/ (- 1 (sqrt 5)) 2) n)))

(define (fibapprox n)  (\( \star \) (/ 1 (sqrt 5)) (expt (/ (+ 1 (sqrt 5)) 2) n)))

```plaintext
>>> (fib 3) 2.0
>>> (fibapprox 3) 1.894427190999916
>>> (fib 5) 5.0
>>> (fibapprox 5) 4.959674775249769
>>> (fib 15) 610.0000000000001
>>> (fibapprox 15) 609.9996721309714
```

Let \( \ell = \frac{1 + \sqrt{5}}{2} \). \( f \) \( n \geq F_n \leq f^{n-1} \)  \( f = 1.6180339887 \)

What if characteristic roots contain complex numbers?

**Ex:** \( t_n = 4t_{n-1} - 5t_{n-2} \) \( t_0 = 2, t_1 = 6 \)

Solution -> \( t_n = (1-i) \cdot (2+i)^n + (1+i) \cdot (2-i)^n \) which yields the correct sequence of values 2, 6, 14, 26, 34, 6, -146, -614, -1726, -3834, ...

What if there are multiple characteristic roots?

**Ex:** \( t_n = 5t_{n-1} + 8t_{n-2} - 4t_{n-3} = 0 \), \( t_0 = 0, t_1 = 1, t_2 = 2 \)

C.E. \( x^3 - 5x^2 + 8x - 4 = 0 \)

\( (x-1)(x-2)^2 = 0 \) roots: 1 (of multiplicity 1), 2 (of multiplicity 2)
Theorem: Given recurrence $c_0 t_n + c_1 t_{n-1} + \ldots + c_k t_{n-k} = 0$, if characteristic roots are $r_1, \ldots, r_m$ of multiplicities $q_1, \ldots, q_m$, then any solution is of the form
\[ \sum_{1 \leq k \leq m} \left( \sum_{0 \leq j = q_k - 1 \atop 0 \leq j} r_k \alpha_j x^j \right). \]

Previous ex.: solution $t_n = \alpha 10n + \alpha 202n + \alpha 21n2n$

\begin{align*}
t_0 &= 0 = \alpha 10 + \alpha 20 \\
t_1 &= 1 = \alpha 10 + 2 \alpha 20 + 2 \alpha 21 \\
t_2 &= 2 = \alpha 10 + 4 \alpha 20 + 8 \alpha 21 \\
\alpha 10 &= -2, \quad \alpha 20 = 2, \quad \alpha 21 = -(1/2) \\
t_n &= 2^{n+1} - n2^{n-1} - 2
\end{align*}

Nonhomogeneous linear equations

Ex: A data structure which supports INSERTion and SEARCH in $O_{\text{wc}}(\log n)$ time is the AVL tree. For every node of an AVL tree, the heights of the left subtree and the right subtree differ by $\leq 1$. Total height of AVL-tree of $n$ nodes satisfies $\left\lceil \frac{\log(n+1)}{\log 2} \right\rceil \leq h(n) \leq 1.4402\ldots \log(n+2) - 0.32772\ldots$ (Gonnet 3.4.1.3, pg. 97) Whence the upper bound?

Let $f(h)$ - fewest nodes in an AVL tree of height $h$.

\begin{align*}
f(0) &= 0 \\
f(1) &= 1 \\
f(2) &= 2 \\
f(3) &= 4 \\
f(h) &= f(h-1) + f(h-2) + 1
\end{align*}

The second order recurrence $f(h) = f(h-1) + f(h-2) + 1$, $f(1) = 1$, $f(2) = 2$ is too hard for summing factor, & it's not homogeneous. Can we make it homogeneous? $f(h) - f(h-1) - f(h-2) = 1$

Replacing $h$ by $h+1$: $f(h+1) - f(h) - f(h-1) = 1$

Subtracting top from bottom: $f(h+1) - 2f(h) + f(h-2) = 0$

$x^3 - 2x^2 + 1 = 0 \implies (x^2 - x - 1)(x - 1) = 0$

Roots: $x_1 = \frac{1 + \sqrt{5}}{2}, x_2 = \frac{1 - \sqrt{5}}{2}, x_3 = 1$
\[ f(h) = \alpha \left( \frac{1 + \sqrt{5}}{2} \right)^h + \beta \left( \frac{1 - \sqrt{5}}{2} \right)^h + \gamma h \]

Plugging in to initial conditions yields
\[ \alpha = 1 + \frac{2}{\sqrt{5}}, \beta = 1 - \frac{2}{\sqrt{5}}, \gamma = -1 \]

Asymptotically, \[ f(h) = \alpha \left( \frac{1 + \sqrt{5}}{2} \right)^h \approx 1.91^{h} \]

\[ \log f(h) = \log \alpha \cdot h + \log \left( \frac{1 + \sqrt{5}}{2} \right)^h + O(1) \approx 1.4402 \log f(h) \]

**Ex:** \( t_n - 2t_{n-1} = 3^n \)

To remove \( 3^n \), replace \( n \) by \( n+1 \) \( \implies t_{n+1} - 2t_n = 3^{n+1} \)

Multiply original by 3 \( \implies 3t_n - 6t_{n-1} = 3^{n+1} \)

Subtracting yields \( t_{n+1} - 5t_n + 6t_{n-1} = 0 \)

C.E. \( x^2 - 5x + 6 = 0 \)

\( (x-2)(x-3) = 0 \)

(Note: 2 root of original equation, 3 from driving function)

**Rule:** Given recurrence of the form

\[ c_0 t_n + c_1 t_{n-1} + \ldots + c_k t_{n-k} = b \cdot n \cdot P(n) \]

- \( b \) is a constant, and,
- \( P(n) \) is a polynomial in \( n \) of degree \( d \),

use characteristic equation \( (c_0 x^k + c_1 x^{k-1} + \ldots + c_k)(x-b)^{d+1} = 0 \)

**Ex:** \( t_n - 2t_{n-1} = (n+5)^*3^n \)

C.E. \( (x-2)^*(x-3)^2 = 0 \)

General solution: \( t_n = \alpha 2^n + \beta 3^n + \gamma n3^n \)

**Domain & Range Transformations**

Back to **Ex:** \( M(n) = 3M \left( \frac{n}{2} \right) + c \cdot n \) (recurrence)

How does one derive that \( M(n) = \Omega(n1.58496...) \)?

\( M(n) - 3M(n/2) = n \)

Replace \( n \) by \( 2^k \) \( \implies M(2^k) - 3M(2^{k-1}) = 2^k \) (so \( k = \log n \))

Let \( t_k = M(2^k) \) \( \implies t_k - 3t_{k-1} = 2^k \) (secondary recurrence)

C.E. \( (x-3)^*(x-2) = 0 \)

General solution: \( t_k = \alpha 3^k + \beta 2^k \implies \text{asymptotically } t_k = \alpha \cdot 3^k \)

\( M(n) = M(2\log n) = \alpha 3\log n + \beta 2\log n = \Omega(3\log n) = O(n\log 3) = O(n1.58496...) \)
**Ex:** Binary search

```haskell
function search(key: typekey; var r.dataarray : integer; 
var hi, lo, mid : integer;
begin lo:=0; hi:=n;
    while hi-lo>0 do begin
    mid:=(hi+lo) div 2;
        if key <= r[mid] then hi:=mid else lo:=mid end;
    if r[hi]=key then search:=hi else search:=-1
end;
```

**Analysis:**

\[ A_n = \begin{cases} 1 & \text{if } n = 1 \\ A_{n/2} + 1 & \text{else} \end{cases} \]

Replace \( n \) by \( 2^k \) (so \( k = \log n \)) - \( A_{2^k} = A_{2^{k-1}} + 1 \)

\[ t_k \leq t_{k-1} + 1 \quad \text{where} \quad t_k = A_{2^k} = A_n \]

\[ t_k \leq k \]

\[ A_{2^k} = A_n = k = \log n \]

**Ex:** Merge Sorting

\[ M(n) = 2M\left(\frac{n}{2}\right) + n - 1, \quad n > 1, \quad M(1) = 0 \]

Let \( n = 2^k \), and \( t_k = M(n) = M(2^k) \)

\[ t_k = 2t_{k-1} + 2^k - 1, \quad k \geq 1, \quad t_0 = 0 \]

\[ t_k - 2t_{k-1} = 2^k - 1 \]

Problem: right side can't be expressed as \( b^n P(n) \)

**New rule:** Given recurrence of the form

\[ c_0 t_n + c_1 t_{n-1} + ... + c_k t_{n-k} = b_1 n P_1(n) + b_2 n P_2(n) + ... \]

- \( b_i \) are constants, and,
- \( P_i(n) \) are polynomials in \( n \) of degree \( d_i \),

use characteristic equation 

\[ (c_0 x^k + c_1 x^{k-1} + ... + c_k)(x - b_1)^{d_1+1}(x - b_2)^{d_2+1} = 0 \]

C.E. \((x-2)^*(x-2)^*(x+1) = (x-2)^2*(x-1) = 0\)

Solution:

\[ t_k = \alpha \cdot 2^k + b k 2^k + \gamma \]

\[ 1^k = \alpha \cdot 2^k + \beta \cdot k 2^k + \gamma \]

\[ t_0 = 0 = \alpha \quad + \gamma \quad \Rightarrow \alpha = -\gamma \]

\[ t_1 = 1 = 2 \alpha \quad + 2\beta \quad + \gamma \quad \Rightarrow \quad 1 = 2\beta \quad - \gamma \quad \Rightarrow \gamma = 2\beta \quad - 1, \quad \alpha = 1 - 2\beta \]

\[ t_2 = 5 = 4\alpha \quad + 8\beta \quad + \gamma \quad \Rightarrow \quad 5 = 4(1 - 2\beta) + 8\beta + 2\beta - 1 = 2\beta + 3 \]

\[ \beta = 1, \quad \alpha = 1 - 1, \quad \gamma = 1, \quad t_k = k 2^k - 2^k + 1 = (k-1)2^k + 1 \]

\[ M(n) = M(2^k) = (\log n-1)n + 1 \]

**Ex:** \( T(n) = 2T\left(\sqrt{n}\right) + \log n \)

Replace \( n \) by \( 2^k \) (so \( k = \log n \))

\[ T(2^k) = 2T\left(2^{k^2}\right) + k \]
Rename \( S(k) = T(2^k) \)
\( S(k) = 2S\left(\frac{k}{2}\right) + k \)

Replace \( k \) by \( 2^j \), and and \( t_j = S(k) = M(2^j) \) (so \( j = \lg k \))
\( t_j - 2t_{j-1} = 2^j \)

\( C.E. (x-2)^*(x-2) = (x-2)^2 = 0 \)

Solution: \( t_j = \alpha \cdot 2^j + \beta \cdot j \) \( 2^j \) (initial conditions yield \( \alpha = 0, \beta = 1 \))
\( t_j = j \cdot 2^j \) \( \implies S(k) = \lg k \cdot (2^l \lg k) = k \cdot \lg k \) \( \implies T(n) = \lg n \cdot \lg \lg n \)

\textbf{Master Theorem} (for divide-and-conquer algorithms):
Given integer constants \( n \geq 1, b \geq 2, k \geq 0 \), and positive real constants \( a \), and \( c \), if \( T : N \to N^+ \) is an eventually nondecreasing function such that
\( T(n) = aT\left(\frac{n}{b}\right) + cn^k \), for \( n > n_0 \)

where \( \frac{n}{n_0} \) is a power of \( b \), then
\[ T(n) = \begin{cases} 
\Theta(n^k) & \text{if } a < b^k \\
\Theta(n^k \log n) & \text{if } a = b^k \\
\Theta(n^{\log_b a}) & \text{if } a > b^k 
\end{cases} \]

\[ \text{--- Full History Recurrences} \]

\textbf{Ex: QuickSort}

\begin{table}[h]
\begin{tabular}{|c|c|c|}
\hline
\(< \mathcal{R} [k] > \) & \(< \mathcal{R} [k] > \) & \(< \mathcal{R} [k] > \) \\
\hline
\hline
\end{tabular}
\end{table}

Initial invocation: \( \text{sort } (r, 1, n) \)

Let \( T(n) \) - expected time to \textit{QuickSort} an array of \( n \) elements, assuming
-every permutation of \( r \) equally likely, and,
elements of \( r \) distinct
\( T(n \mid \text{pivot ends at } k) = n + T(k - 1) + T(n - k) \)

\( \Pr\{\text{pivot ends at } k\} = \frac{1}{n}, 1 \leq k \leq n \)

\( T(n) = \sum_{1 \leq k \leq n} [n + T(k - 1) + T(n - k)] \Pr\{\text{pivot at } k\} = \frac{1}{n} \sum_{1 \leq k \leq n} [n + T(k - 1) + T(n - k)] \)
\[= n + \frac{1}{n} \sum_{1 \leq k \leq n} [T(k-1) + T(n-k)] = n + \frac{2}{n} \sum_{1 \leq k \leq n} T(k-1)\]
\[n \cdot T(n) = n^2 + 2 \sum_{1 \leq k \leq n} T(k-1) = n^2 + 2T(n-1) + 2 \sum_{1 \leq k \leq n-1} T(k-1) \quad (*)\]

Replace \(n\) by \(n-1\),
\[(n-1) \cdot T(n-1) = (n-1)^2 + 2 \sum_{1 \leq k \leq n-1} T(k-1) \quad (**)

Subtracting (**) from (*),
\[n \cdot T(n) - (n-1) \cdot T(n-1) = 2n - 1 + 2T(n-1)\]
\[T(n) = \frac{n+1}{n}T(n-1) + 2 - \frac{1}{n}\]

Using summing factors,
\[T(n) = \frac{(n+1)!}{n!} \sum_{1 \leq k \leq n} \frac{2 - \frac{1}{k}}{k} = 2(n+1)H_n + O(1)\]

**Ex:** Binary tree search

Consider randomly grown binary tree. What is \(E[A_n]\)?
-every permutation equiprobable, i.e., \(Pr\{\text{permutation}\} = 1/n!\)
\(E[A_n]\) - average internal path length - average length of path from root to a node

Let \(P_n\) - r.v. denoting total path length to all nodes
For given root \(i\), \(1 \leq i \leq n\), let \(P_L\) - total path length of left subtree
\(P_R\) - total path length of right subtree
\[P_n = 1 + P_L + P_R + (n-1) = n + P_L + P_R = n + P_{i-1} + P_{n-i}\]
root

\(\text{~each nonroot node must pass through root}~\)
\[E[P_n] = \frac{1}{n} \sum_{1 \leq k \leq n} (n + E[P_{k-1}] + E[P_{n-k}]) = n + \frac{1}{n} \sum_{1 \leq k \leq n} E[P_{k-1}] + \frac{1}{n} \sum_{1 \leq k \leq n} E[P_{n-k}]\]

Replacing \(k\) in 2 nd term by \(k+1\), & in 3 rd by \(n-k\),
\[E[P_n] = n + \frac{1}{n} \sum_{1 \leq k+1 \leq n} E[P_{k+1}] + \frac{1}{n} \sum_{1 \leq n-k \leq n} E[P_{n-k}] = n + \frac{1}{n} \sum_{0 \leq k \leq n-1} E[P_{k}] + \frac{1}{n} \sum_{0 \leq k \leq n-1} E[P_{n-k}] = n + \frac{2}{n} \sum_{0 \leq k \leq n-1} E[P_{k}]\]
\[ n \cdot \mathbb{E}[P_n] = n^2 + 2 \sum_{0 \leq k \leq n-1} \mathbb{E}[P_k] = n^2 + \sum_{0 \leq k \leq n-1} \mathbb{E}[P_k] + 2\mathbb{E}[P_{n-1}] \]

\[(n - 1)\mathbb{E}[P_{n-1}] = (n - 1)^2 + 2 \sum_{0 \leq k \leq n-1} \mathbb{E}[P_k] \]

Subtracting last eq’n from preceding,
\[ n^*\mathbb{E}[P_n] - (n - 1)^*\mathbb{E}[P_{n-1}] = n^2 - (n - 1)^2 + 2\mathbb{E}[P_{n-1}] \]

\[ n^*\mathbb{E}[P_n] = 2n - 1 + (n + 1)^*\mathbb{E}[P_{n-1}] \]

\[ \mathbb{E}[P_n] = \frac{2n - 1}{n} + \frac{n + 1}{n} \mathbb{E}[P_{n-1}] \]

Using summing factors, where \[ c_n = \frac{2n - 1}{n}, b_n = \frac{n + 1}{n}, \text{ and noting that } \prod_{1 \leq k \leq n} b_k = n + 1, \]
\[ \mathbb{E}[P_n] = (n + 1) \sum_{1 \leq k \leq n} \frac{2k - 1}{k(k + 1)} = 2(n + 1) \sum_{1 \leq k \leq n} \frac{1}{k + 1} - (n + 1) \sum_{1 \leq k \leq n} \frac{1}{k(k + 1)} \]
\[ = 2(n + 1)(H_{n+1} - 1) - (n + 1) \sum_{1 \leq k \leq n} \frac{1}{k(k + 1)} \]

Note that \[ \frac{1}{k(k + 1)} = \frac{1}{k} - \frac{1}{k + 1} \]
\[ \mathbb{E}[P_n] = 2(n + 1)(H_{n+1} - 1) - (n + 1)(H_n - H_{n+1} + 1) \]
\[ = 2(n + 1)(H_{n+1} - 1) - (n + 1) \left( 1 - \frac{1}{n + 1} \right) = 2(n + 1)(H_{n+1} - 1) - n \]

(Gonnet, 3.4.1.1, pg. 94)
\[ \mathbb{E}[A_n] = \frac{\mathbb{E}[P_n]}{n} = 2 \left( 1 + \frac{1}{n} \right)(H_{n+1} - 1) - 1 = 2 \left( 1 + \frac{1}{n} \right)H_n - \frac{2}{n^2} - 3 \]