374 10/4/22

Divide and conquer continued

Kartsuba's algorithm

$(\alpha x+\beta)(\gamma x+\delta) = \alpha\gamma x^2 + \alpha\delta x + \beta\gamma x + \beta\delta$

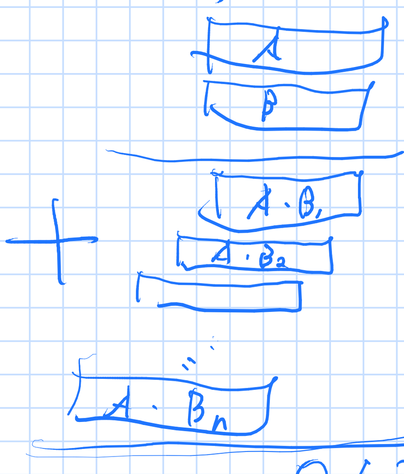
$= \boxed{\alpha\gamma} x^2 + \boxed{(\alpha\delta + \beta\gamma)} x + \beta\cdot\delta$ 4

$(\alpha+\beta)(\gamma+\delta) = \boxed{\alpha\gamma} + \boxed{\alpha\delta+\beta\gamma} + \boxed{\beta\delta}$

$\alpha\delta+\beta\gamma = \underbrace{(\alpha+\beta)(\gamma+\delta)}_{1} - \underbrace{\alpha\gamma}_{2} - \underbrace{\beta\delta}_{3}$

$\underbrace{7341\cdots\cdots 7}_{n} \times \underbrace{31\cdots\cdots 2996}_{n}$

$O(n^2)$ standard primary school algorithm



$O(n^2)$

$O(n^2)$ algorithm

123456

$123x + 456$ $\quad x = 10^3$

$\boxed{a_n a_{n-1} \quad a_0}$

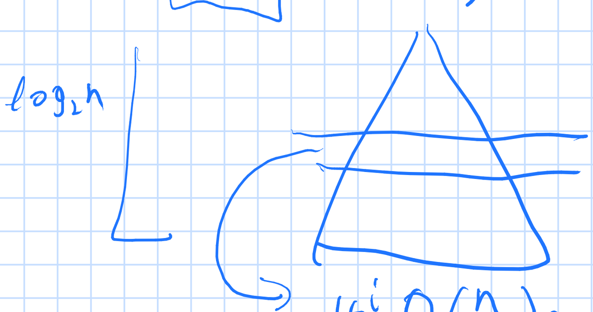
$\boxed{a_n a_{n-1} \quad a_{n/2}}\, x + \boxed{a_{n/2-1} a_{n/2-2} \cdots a_1}$ $\quad x^{n/2}$

$A = A_L x + A_R$ $\quad n = |A| \quad |A_L| = |A_R|$

$B = B_L x + B_R$ $\quad \boxed{x = 10^{n/2}}$

$A\cdot B = A_L B_L x^2 + (A_L B_R + A_R B_L)x + A_R B_R$ $\quad$ 4 multiplications, $n/2$ digits

$T(n) = O(n) + 4T\left(\frac{n}{2}\right) \Rightarrow O(n^2)$



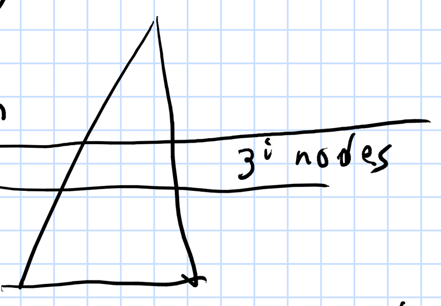
4^i nodes $\quad \frac{n}{2^i}$

$4^i\, O\left(\frac{n}{2^i}\right) = O(2^i n)$

$\sum_{i=0}^{\log_2 n} O(2^i n) = O(n) \sum_{i=0}^{\log_2 n} 2^i = O(n)\cdot O(n) = O(n^2).$

$T(n) = 3T\left(\frac{n}{2}\right) + O(n)$

Total work in ith level is $3^i \frac{n}{2^i} = \left(\frac{3}{2}\right)^i n$



$\frac{n}{2^i}$

$T(n) = \sum_{i=0}^{\log_2 n} O\left(\left(\frac{3}{2}\right)^i n\right) = O(n) \underbrace{\sum_{i=0}^{\log_2 n} \left(\frac{3}{2}\right)^i}$

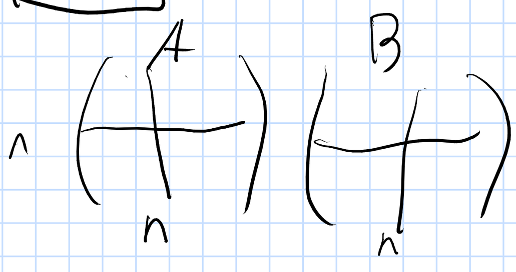
$= O\left(n\cdot\left(\frac{3}{2}\right)^{\log_2 n}\right) = n\left(2^{(\log \frac{3}{2})}\right)^{\log_2 n}$

$= n \cdot 2^{\log\frac{3}{2}\cdot \log n}$

$= n\left(2^{\log_2 n}\right)^{\log_2 3/2}$

$= n\cdot n^{\log_2 3/2} = O\left(n^{1+\log_2 \frac{3}{2}}\right)$

$\boxed{\text{FFT}}$



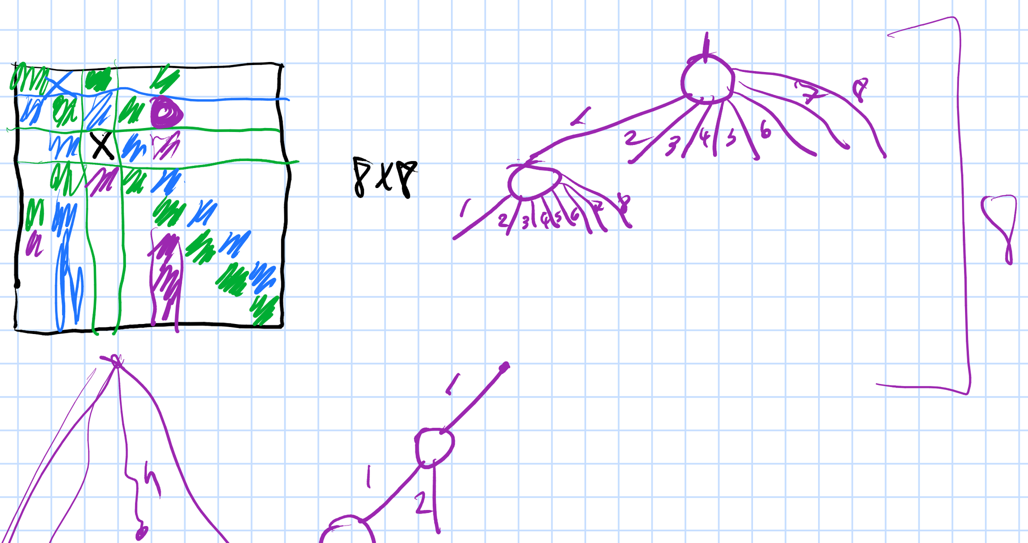
$O(n^3)$ $\quad$ strassen algorithm

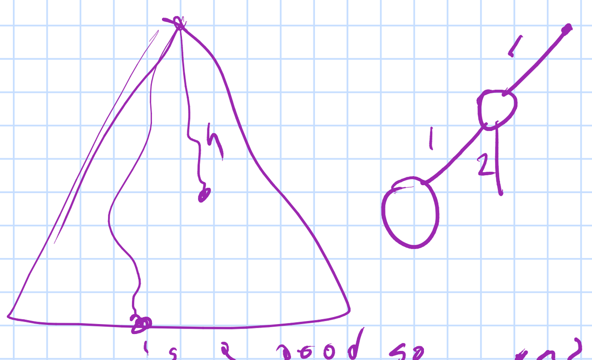
$8 \Rightarrow 7$

$O(n^{2.41\ldots})$

Search problems

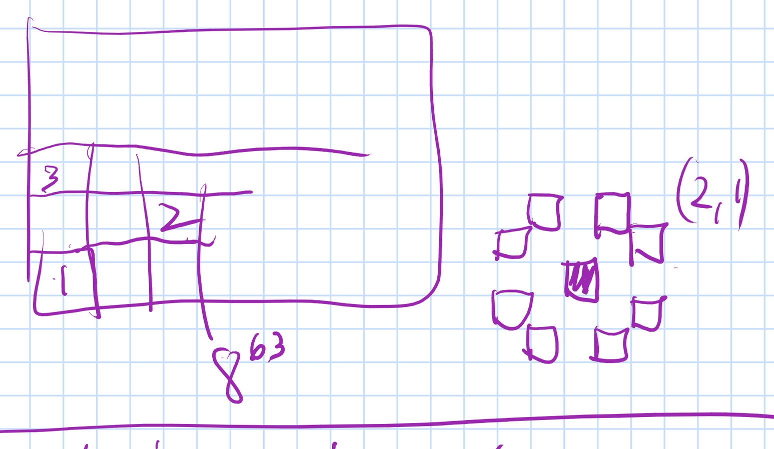
8 queens





is a good so on?

backtracking

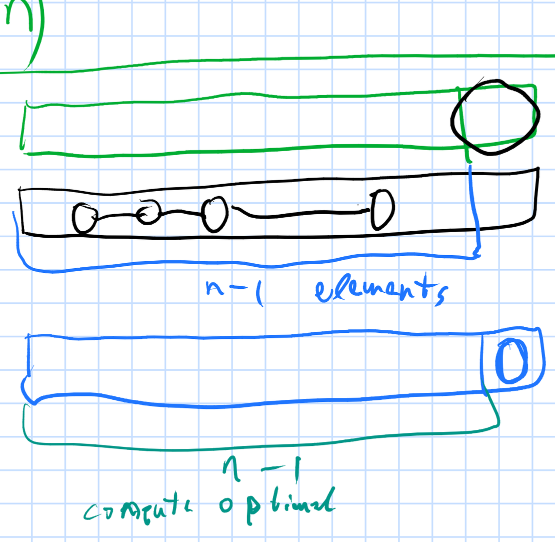


Longest increasing subsequence

A[1...n]: Array with n numbers.

7	1	13	26	-4	2.8	17
0	1	0	0	1	0	0

$O(2^n n)$



A[n] is not in the optimal solution

A[n] is in the optimal solution

$\underbrace{3\ 7}$ 4

$f(i,x)$ = longest increasing subseq in A[1...i] s.t. all the numbers are smaller than x.

$$f(i,x) = \begin{cases} 0 & i=0 \\ f(i-1,x) & A[i] > x \\ \max\begin{pmatrix} f(i-1, A[i])+1 \\ f(i-1,x)\end{pmatrix} & \text{otherwise } A[i] < x \end{cases}$$

$f(n,+\infty) \equiv$ longest subsequence

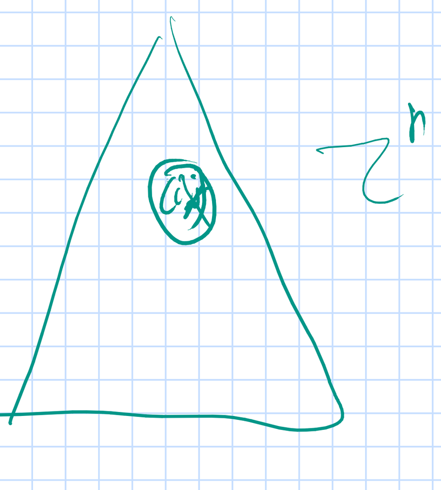
$f(i,x)$: $\quad i \in \{0,1,2,\ldots n\}$ $\quad n+1$

$x \in \{A[1], A[2], \ldots, A[n], +\infty\}$ $\quad n+1$

A[1....n] given

```
LIS(i, x):
    if i=0 then return 0
    if A[i]>x then return LIS(i-1,x)
    α ← LIS(i-1, x)
    β ← LIS(i-1, A[i])+1
    return max(α, β)
```

$\boxed{\text{LIS}(n,+\infty)}$



$O(n^2)$ different values