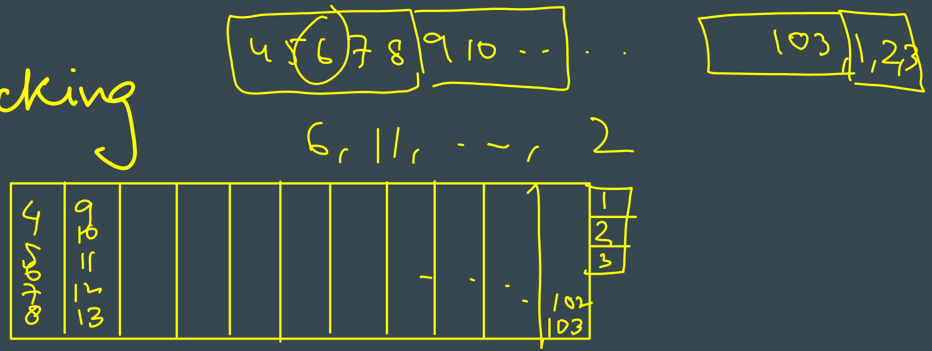
Thursday – Backtracking

Tuesday – FFT




# CSE525 Lec6: Backtracking

...

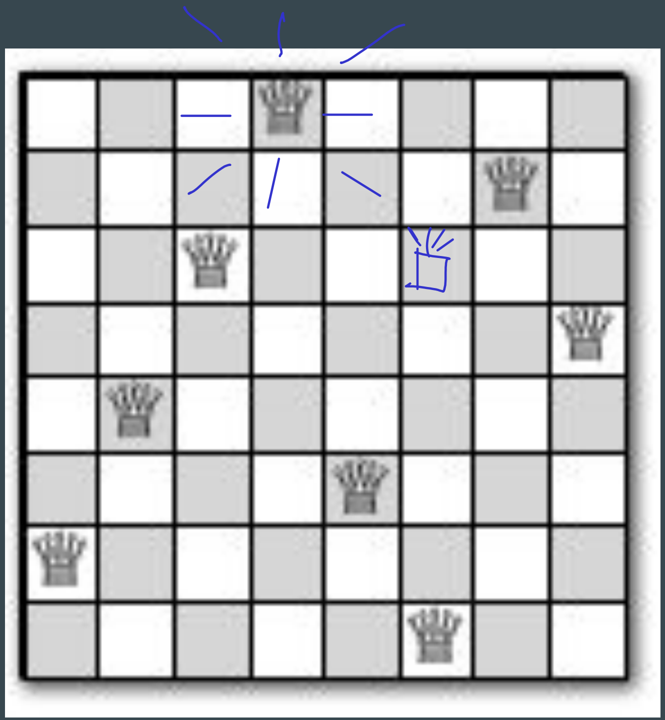
$DFT(A(x), K) \rightarrow [B_1, \ldots B_K]$

$B_j = A(\omega_K^j)$

rank(K) ≥ ___32___ ?

Debajyoti Bera (M21)

# Eight-Queens problem



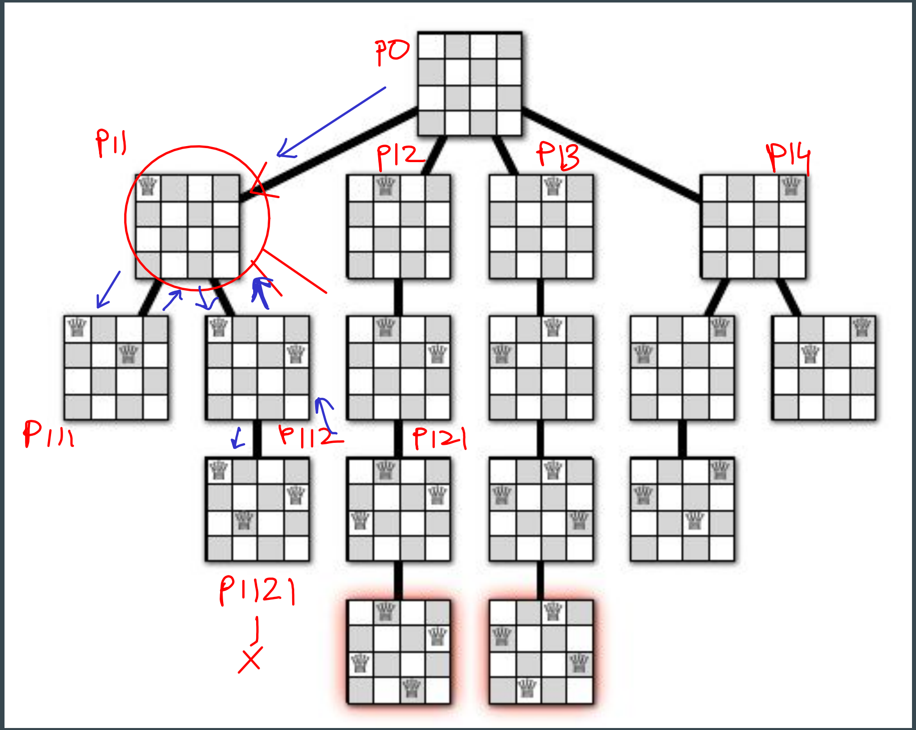
Search for a feasible solution among <u>all solutions/configurations</u>.

Make all possible choices →

1. Suppose we make a local choice towards a solution.
2. Can the $\underbrace{\text{restricted problem}}_{\text{subproblem}}$ be solved recursively

1+2.

# Eight-Queens problem




Search for a feasible solution among all solutions/configurations by trying all possible next-steps.

acyclic directed graph

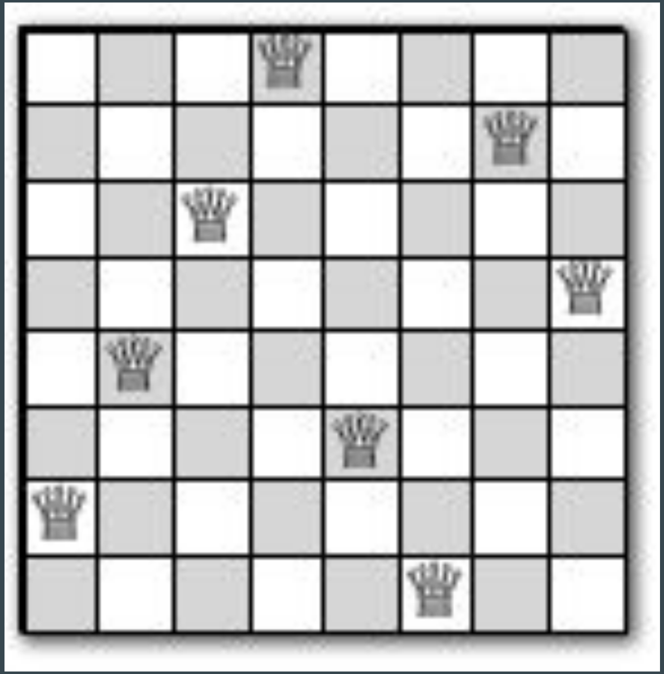
- Systematically explore all configs.
    - Use recursion
    - Incrementally build a feasible solution
    - Prune/discontinue bad configurations

PlaceQueens $Q = [2, 7, 1, 0, 0, 0, 0, 0]$ $i=4$

$\omega_8^{2\cdot 0}$ . . . $\omega_8^{2\cdot 3}$ $\omega_8^{2\cdot 4}$ $\omega_8^{2\cdot 5}$ . . . $\omega_8^{2\cdot 7}$

$\omega_4^0$ $\omega_4^1$ . . . $\omega_4^3$ $\omega_4^4 = \omega_4^0$ $\omega_4^5 = \omega_4^1$

# Eight-Queens problem



// where to place queens in rows i ... n given a partial placement of queens in rows 1 ... (i-1)

PlaceQueens(Q[1 ... n], i):

    For all column c:
        If (i,c) is a feasible position:  exercise
            Q[i] = c
            PlaceQueens(Q,i+1)
        Else
            // No need to consider

$\omega_8^{2j} = e^{2\pi i \cdot 2j/8} = e^{2\pi i \cdot j/4} = \omega_4^j$

$\omega_{2n}^{2j} = \omega_n^j$

$= e^{2\pi i \cdot 2j/2n} = e^{2\pi i \cdot j/n} = \omega_n^j$

**Q.** Where to place queens on an empty board? List all solutions. $T(k) = n * T(k-1) + \text{Time(feasibility)} * n$

**Q.** How to determine running time for n x n board ?

$T(k)$ placing k queens in nxn board where 1st n-k rows have queens ∴ $T(n)$ = complexity for entire board

# Subset-Sum

X = {8,17,6,5,3,10,9}  T=15 Valid SS exists

X = {11,6,5,1,7,13,12}  T=15 No SS exists

Idea: Search among all subsets of X.

- Systematically
- Recursively
- Incrementally

- Solution contains 8
  - Solution contains 17
    - ???
  - Solution does not contain 17
    - ???
- Solution does not contain 8
  - Solution contains 17
    - ???
  - Solution does not contain 17
    - ???

X' = {17, 6,5,3,10,9},
T= 7

X' = {17,6,5,3,10,9}
T = 15

# Subset-Sum

$T(n,k)$: worst case time complexity of ISS on an n-sized X and k-bit number T

$\leq T(n-1,k) + T(n-1,k) + O(1) = 2T(n-1,k) + O(1)$
$= O(2^n)$

// return True iff A has a subset with sum t

Write recursive code for SS(A,t)

// Assume that A is a global array and only

// pass left/right indices

add base case

IsSubsetSum(X,T):
  e: first element of X
  choice1 = IsSubsetSum(X-e, T-e) ← $T(n-1,k)$
  choice2 = IsSubsetSum(X-e, T) ← $T(n-1,k)$
  return choice1 ∨ choice2

eg. T=15, e=1

#bits to represent T-e → as large as T

**Q: Discuss its time-complexity**

$T(n)$: worst case time complexity of IsSubsetSum on an n-sized X

$= 2T(n-1) + O(1) = O(2^n)$

# Longest Increasing Subsequence

BENT SQUARING ... are subsequences of SUBSEQUENCEBACKTRACKING

LIS

**Q.** Given integer array A[1 ... n], find the length of its longest increasing subsequence.

Incremental Recursive Systematic

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 1 | 4 | 1 | 5 | 9 | 2 | 6 | 5 | 3 | 5 | 8 |

Try to find the correct solution by looking at all increasing subsequences that start with 3, and that do not start with 3. Try to find this recursively.

LIS( A[2 ... n])

Suppose we know that LIS does not start with 3. Can you identify it recursively?

LIS(A[2 ... n]) → first element

Suppose we know that LIS starts with 3. Can you identify it recursively?

# Longest Increasing Subsequence

**Q.** Given integer array A[1 … n], find the length of its longest <u>increasing</u> subsequence.

Incremental Recursive Systematic

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 1 | 4 | 1 | 5 | 9 | 2 | 6 | 5 | 3 | 5 | 8 |

**Q.** What <u>problem</u> should be (recursively) solved? Assume that A is global.

LIS(i) =

- Given input i, return LIS of A[i … n]

# Longest Increasing Subsequence

**Q.** Given integer array A[1 ... n], find the length of its longest increasing subsequence.

Incremental Recursive Systematic

| 3 | 1 | 7 | 5 | 4 | 9 | 2 | 6 | 5 | 3 | 5 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

**Q.** What problem should be (recursively) solved? Assume that A is global.

- LIS1(**prev,i**): Given input prev and i > prev, return LIS of A[i ... n] in which every element is larger than **A[prev]**

LIS1 ( 4, 9) = LIS of A[9... 12] in which every elmt of the subseq. > A[4] = 5
prev i
= 1 [8]

A[7, ... ]

**LIS1(1,7) = ???**

exhaustive

**Suppose ... A[1] >= A[7]** = LIS1(1,8)

**Suppose ... A[1] < A[7], LIS1(1,8) = 2 and LIS1(7,8) = 3**

element A[7]

Subseq has A(7)

Can you find the subsequence itself? = max (1+LIS1(7,8), LIS1(1,8))

local choices are exhaustive 1 + LIS1(1,8) X subseq doesn't have A(7)

# Longest Increasing Subsequence

**Q.** Given integer array A[1 ... n], find the length of its longest <u>increasing</u> subsequence.

Incremental Recursive Systematic

| 3 | 1 | 7 | 5 | 4 | 9 | 2 | 6 | 5 | 3 | 5 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

**Q.** What <u>problem</u> should be (recursively) solved? Assume that A is global.

- LIS1(**prev,i**): Given input prev and i > prev, return LIS of A[i ... n] in which every element is larger than **A[prev]**

**LIS1(prev,i) =** (recursive formula)

Can you find the subsequence itself?

$$LIS(A[1 \cdots n]) = \max_{i=1\cdots n} LIS2(i)$$

# Longest Increasing Subsequence

**Q.** Given integer array A[1 ... n], find the length of its longest increasing subsequence.

Incremental Recursive Systematic

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 1 | 4 | 1 | 5 | 9 | 2 | 6 | 5 | 3 | 5 | 8 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

**Q.** What problem should be (recursively) solved? Assume that A is global.

- **LIS2(i)** : Given input i, return LIS of A[i ... n] in which the subsequence starts with **A[i]**

$LIS2(8) = 2$ [6, 8]

A[5 ⋯] that starts with 5

A[3 ⋯ ]

**LIS2(3) = ???** $1 + \max_{i > 3,\ A[3] < A[i]} LIS2(i) = \max\{LIS2(5), LIS2(6), LIS2(8), LIS2(9), LIS2(11), LIS2(12)\}$

3,

6 local choices - element after A[3] (exhaustive)

**Suppose ... LIS2(4) = 3**