Lec-7
DSA-II 5-10-PM

① abc → Permutation

① N = 4

N-queen T → Dollar

$^{N}C_{N}$?!

$^{3}C_{2}$ = 3

$T_q = 2$

$^{n}C_{r}$?! = $\frac{n!}{(n-r)!\,r!}$

$^{n}P_{r}$ = $\frac{n!}{(n-r)!}$

$T_q = 2$

Backtrack

$T_q = 2$

Base Point

Ind base

C++

$T_q = 2$

```
public static void Permutation(boolean[] board, int tq, int qpsf, String ans) {
    if (qpsf == tq) {
        System.out.println(ans);
        return;
    }
    for (int i = 0; i < board.length; i++) {
        if (board[i] == false) {
            board[i] = true;// Queen Placed
            Permutation(board, tq, qpsf + 1, ans + "b" + i + "q" + qpsf);
            board[i] = false;// Queen removed
        }
    }
}
```

```
public static void Permutation(boolean[] board, int tq, int qpsf, String ans) {
    if (qpsf == tq) {
        System.out.println(ans);
        return;
    }
    for (int i = 0; i < board.length; i++) {
        if (board[i] == false) {
            board[i] = true;// Queen Placed
            Permutation(board, tq, qpsf + 1, ans + "b" + i + "q" + qpsf);
            board[i] = false;// Queen removed
        }
    }
}
```

① $T_q = 2$

Combination

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    int n = 4;
    boolean[] board = new boolean[n];// false
    int[] ans = {};
    Combination(board, tq, 0, "",0);
}

public static void Combination(boolean[] board, int tq, int qpsf, String ans,int idx) {
    if (qpsf == tq) {
        System.out.println(ans);
        return;
    }
    for (int i = idx; i < board.length; i++) {
        if (board[i] == false) {
            board[i] = true;// Queen Placed
            Combination(board, tq, qpsf + 1, ans + "b" + i + "q" + qpsf,i+1);
            board[i] = false;// Queen removed
        }
    }
}
```

[m, i, l, i, n]

[m, i, l, n]

[m, i, n]

[m i n]

[m, i, l, l, n]

m i l l n

m i l l n

m i l l n

o b c d

o | b c d

o | b | c d

o | b | c | d

o b c | d

o b c | d

o b c d

Pop(&(List<Character>) cutx)

Pop(&(List<Character>) cutx)

Pop(&(List<Character>) cutx)

Pop(&(List<Character>) cutx)

Pop(&(List<Character>) cutx)

Pop(&(List<Character>) cutx)

Pop(&(List<Character>) cutx)

Pop(&(List<Character>) cutx)

① Check

Mississippi

Mississippi

Mississippi