Subsequences and subarrays

1. جميع الـ Subarrays (متتالية):

```
void Subarrays(const vector<int>& arr) {
    int n = arr.size();
    for (int i = 0; i < n; ++i) {
        vector<int> sub;
        for (int j = i; j < n; ++j) {
            sub.push_back(arr[j]);
        }
    }
}
```

- عدد الـ subarrays هو $n * (n+1)/2$ <-- O(n^2)

2. جميع الـ Subsequences (مش شرط متتالية):

```
void Subsequences(const vector<int>& arr) {
    int n = arr.size();
    for (int mask = 1; mask < (1 << n); ++mask) {
        vector<int> subseq;
        for (int i = 0; i < n; ++i) {
            if ((mask >> i) & 1)
                subseq.push_back(arr[i]);
        }
    }
}
```

- لأن عدد الـ 2^n <-- O(2^n)

Expected Value – Quick Contest Note

Expected Value = sum of (value × probability)

General Formula:

$E[X] = \sum(value_i \times prob_i)$

Quick Code (Uniform Case): sum of values / n

```
int expected = 0;
for (int i = 0; i < n; i++)
    expected = (expected + a[i]) % MOD;

expected = 1LL * expected * modInverse(n) % MOD;
```

If selecting k elements out of n:

- Total subsets = `nCr(n, k)` - Expected contribution of element `i`:
  Often → count = `nCr(i, k-1)` for each element
  Final Expected Value (Weighted Sum):

```
ans = total_contribution × modInverse(nCr(n, k)) % MOD;
```