subarray → [0, 1, 2, 4]   (subarray)

[0], [0,1], [0,1,2], [0,1,4] $^nC_2$

[0,2] α

n array → $\left(\frac{n \times (n+1)}{2}\right)$

→ pick
→ not-pick.

subsequence / subset → [1, 2, 3]

'n' → 2^n subsequences.

[1,2,3]
[1],[2],[3],
[1,2],[1,2,3],
[2,3].

$\frac{3 \times 4 \times 2}{2}$

[1],[2],[3] [ ]
[1,2],[1,3],[2,3],[1,2,3]
sub seq

[1,3] [1,2]

pick, not-pick

(soch)

(array)
(ds) (array)

i=0, | 1 2 3 | |

8

// base case.
(index check)

pick	not-p
o) ds - add	o) ds - no d
o) index incr	o) index inc.

i=1, | 1 | [1,(2)3]

[1,2,3]
1

i=1, |

(new)
(new) (new)

[1,2,3] i=2 | |

i=2, | 2 |

i=2, | i=2, | 2 1 |

[1, 2, (3)]

final all

(final arr)

i=3,   i=3   i=3   i=3 | 3 2 | i=3 | 1 | | 3 1 | | 2 1 | | 3 2 1 |

2d

[ ],[3],[2],[2,3],[1],[1,3],[1,2],[1,2,3]] (specific order)

if (index > n)
{
  final_arr.push_back(new_array);
  return;
}

[ ],[3],[2],
[2,3]

f (int i, new-arr, old-arr
{

}

15.64 MB

[[char]],

8 new

i=2 | 3 | ?

new
pick (   )

α

i=3

| 3 |

i=3 | 3 |

pick

| 3 | new

[[3]]

ih

ddihin

vector<vector<int>>
)final_arr)