Write-up cho bài 8:

Tổng cần tìm có thể được biểu diễn bằng công thức :

https://espresso.codeforces.com/77c03e9ad6d2900d294cf74c0feebc48bcdd9a73.png ( là tổng các tổ hợp C(n,r) với r chạy từ 1-> n nhân với ) (1)

Quá trình để ra được tổng này:

https://espresso.codeforces.com/e01d85c97561e79e2cc4a9b0c589d6b7f5476c54.png (2)

Đạo hàm và nhân với x, ta ra được:

https://espresso.codeforces.com/9c76977f88e610e3d9aae052a7a8038246db653f.png

Ta thử đạo hàm 1 lần nữa và nhân tiếp cho x, sẽ ra:

https://espresso.codeforces.com/4af83424dbfd0a8f99c4d63b9eb4ad1cb64bcff3.png

Lặp lại quá trình này k lần và thay thế x = 1, sẽ ra được tổng cần tính (1)

Để thực hiện được phép tính này, ta sẽ sử dụng quy hoạch động.

Giả sử dp[a][b][c] là giá trị của biểu thức *x* *b*(1 + *x*)*c* sau khi thực hiện đạo hàm và nhân với x, **a lần** với x = 1. Suy ra, như một cách biểu diễn khác cho công thức (2), hàm dp sẽ có dạng là dp[k][0][n]. (vì công thức (2), xb = 1 suy ra b = 0, a lần đạo hàm, vì như trên ta cần đạo hàm k lần nên a = k, c theo công thức (2) = n)

Quay lại với công thức tổng quát https://espresso.codeforces.com/6e8dbc363e59aee1313d02bf938e53c267dc9007.png sau 1 lần đạo hàm và nhân x, sẽ thành:

https://espresso.codeforces.com/4fffd29bb680681a4cfdb76f7f579205141183d9.png = https://espresso.codeforces.com/6660832239eecefe87f52f7babd8476d8a9b616b.png (3)

Vì vậy, công thức truy hồi sẽ là: *dp*[*a*][*b*][*c*] = *b* \* *dp*[*a* - 1][*b*][*c*] + *c* \* *dp*[*a* - 1][*b* + 1][*c* - 1] (thay thế phương trình 3 thành hàm dp[][][] như trên giả sử)

Độ phức tạp thuật toán: O(k2)

Sol: <https://ideone.com/X5REGx>