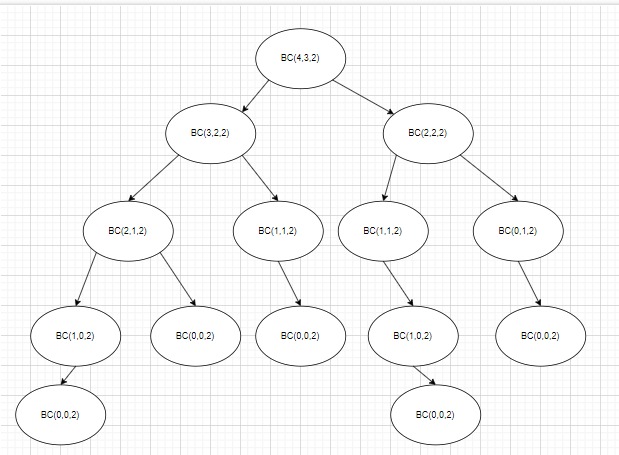
==================================== SKRIP ========================================

1. Penjelasan Problem:

Jadi problem ini tentang barcode. jadi, kita disuruh untuk mencari jumlah barcode berdasarkan input yang dimasukkan. Inputnya terdiri dari 3 bilangan bulat, yaitu n, k, m. n itu maksudnya jumlah unit total dari barcode. k maksudnya adalah banyak bar pada barcode (bisa hitam atau putih). m artinya adalah maksimal ketebalan bar. hal ini bisa kita tuliskan sebagai begini, BC(n, k, m), yaitu himpunan barcode - barcode yang memenuhi syarat. misalnya, kalo kita menerima inputan 7 4 2, maka artinya adalah 7 adalah jumlah unit total yang ada pada barcode, 4 artinya jumlah bar yang ada pada barcode, dan 2 artinya adalah jumlah maksimal ketebalan bar. kalo kita masukkan 7 4 2, ternyata jumlah total barcode yang memenuhi syarat tersebut ialah 4 barcode.

2. Kenapa Termasuk dalam Dynamic Problem.



Masalah ini bisa dipecah – pecah menjadi subproblem – subproblem kecil. Jika dilihat dari gmbr di atas, misalnya inputannya adalah 4 3 2. 4 3 2 ini bisa di pecah menjadi sbproblem – subproblem kecil, yaitu BC(n,k,m)= BC(4, 3, 2) = BC(4 - 1, 2, 2) +… + BC(4-m, 2, 2). Inilah bagian optimal substructurenya. Jika diperhatikan daripada tree berikut, maka masalah ini dapat dipecah – pecah menjadi masalah – masalah yang lebih kecil lagi sampai menuju kepada base casenya, yaitu BC(0,0,2).

Jika tree tersebut diperhatikan lagi, maka bisa terlihat bahwa ada beberpa subproblem yang ternyata sama. Hal ini akan membuat komputer memboroskan resources nya karena harus menghitung problem yg sama berulang kali. Inilah bagian overlappingnya.

Melalui dynamic programming, kita akan mencoba untuk mengurangi pemakaian resources komputer dengan cara menyimpan hasil komputasi yang sudah dilakukan sehingga tidak perlu melakukan perhitungan berulang kali.

3. Tree dan memoization

Berikut saya tampilkan sc saya:

while True:

    try:

        n, k, m = map(int, input().split())

    except:

        break

    dp = [[0]\*(k+1) for i in range(n+1)]

    dp[0][0] = 1

    for i in range(1, n+1):

        for j in range(1, k+1):

            for x in range(1, min(m, i)+1):

                dp[i][j] += dp[i-x][j-1]

    print(dp[n][k])

saya akan mencoba untuk menjelaskan souse code saya secara rinci.

while True:

while true disini adalah tujuannya supaya penggua bisa memasukkan inputannya berkali – kali.

try:

        n, k, m = map(int, input().split())

    except:

        break

nah kemudian, kita meminta inputan oleh pengguna yang akan kita masukkan pada variabel n, k, dan m. jika semisalnya pengguna tidak memasukkan inputtan sama sekali, maka looping akan diberhentikan dan program selesai.

dp = [[0]\*(k+1) for i in range(n+1)]

    dp[0][0] = 1

nah, kita persiapkan array 2d dulu (atau list) dengan ukuran (k+1)(n+1) yang gunanya untuk menyimpan value hasil komputasi , sehingga tidak perlu melakukan perhitungan berkali – kali. Kemudian, kita tetapkan nilai dp[0][0] sebagai base casenya, yaitu 1.

    for i in range(1, n+1):

        for j in range(1, k+1):

            for x in range(1, min(m, i)+1):

                dp[i][j] += dp[i-x][j-1]

    print(dp[n][k])

kemudian, kita akses semua cell yang ada pada array 2d, dimulai daripada [1][1]. Kemudian, kita cari jumlah barcode yang memenuhi syarat, dengan cara menotal semua subproblem yang ada seperti yang sudah dijelaskan sebelumnya. Kemudian, hasilnya disimpan ke dalam dp[i][j].

jika iterasi sudah selesai, maka kemudian kita tinggal mengoutputkan hasil akhirnya