indel = 2   mismatch = 3

TACTGACG

TAGTGTCG

3 3

= 6

TAC_TGACG

TA_GTGTCG

2 2 3

= 7

TAC_TGA_CG

TA_GTG_TCG

2 2 2 2

= 8

$\boxed{S}$ C_1 C_2 C_3 . . . . C_8

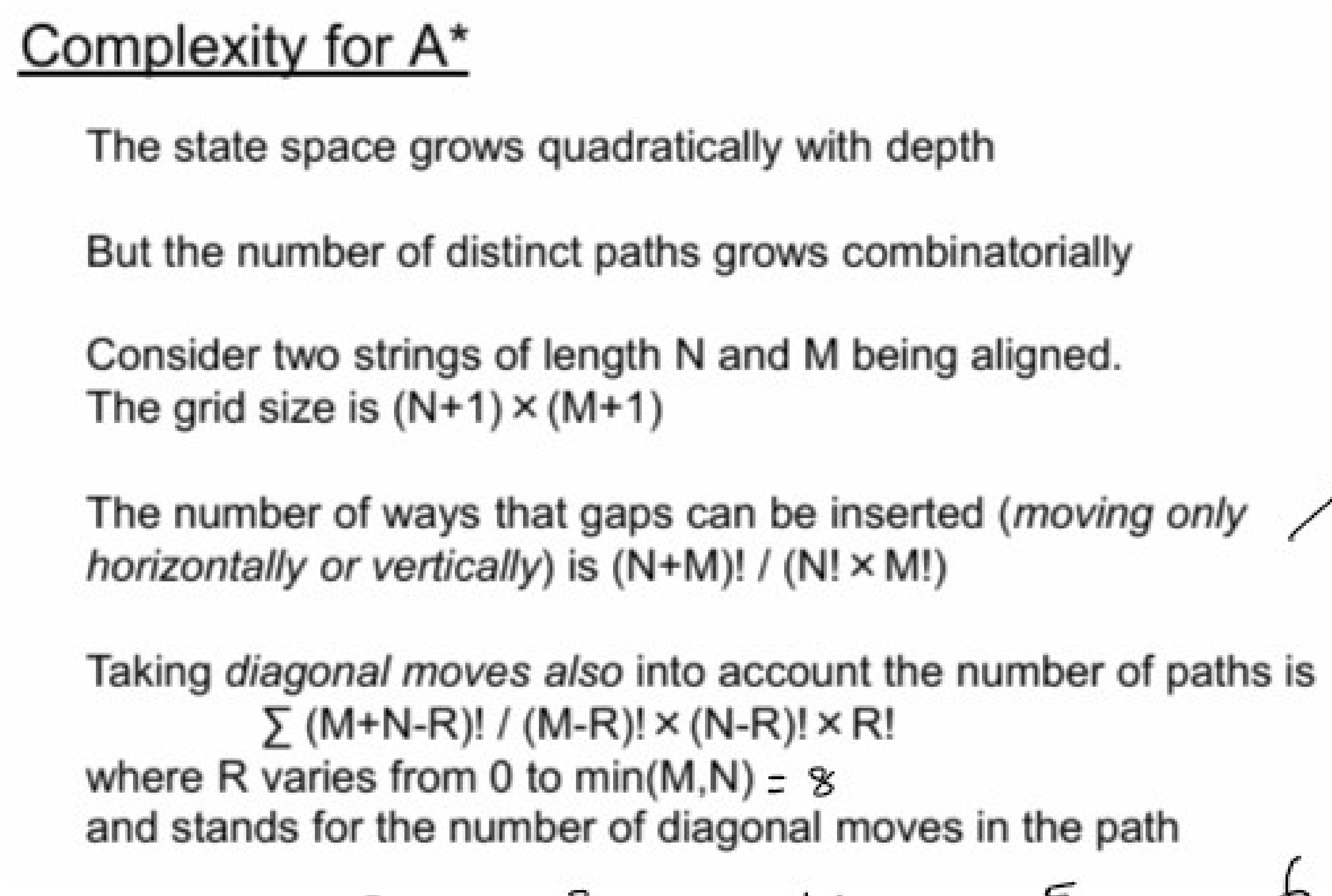
C_1

C_2

⋮

C_8

$\boxed{9}$



Complexity for A*

The state space grows quadratically with depth

But the number of distinct paths grows combinatorially

Consider two strings of length N and M being aligned.
The grid size is (N+1) × (M+1)

The number of ways that gaps can be inserted (*moving only horizontally or vertically*) is (N+M)! / (N! × M!)

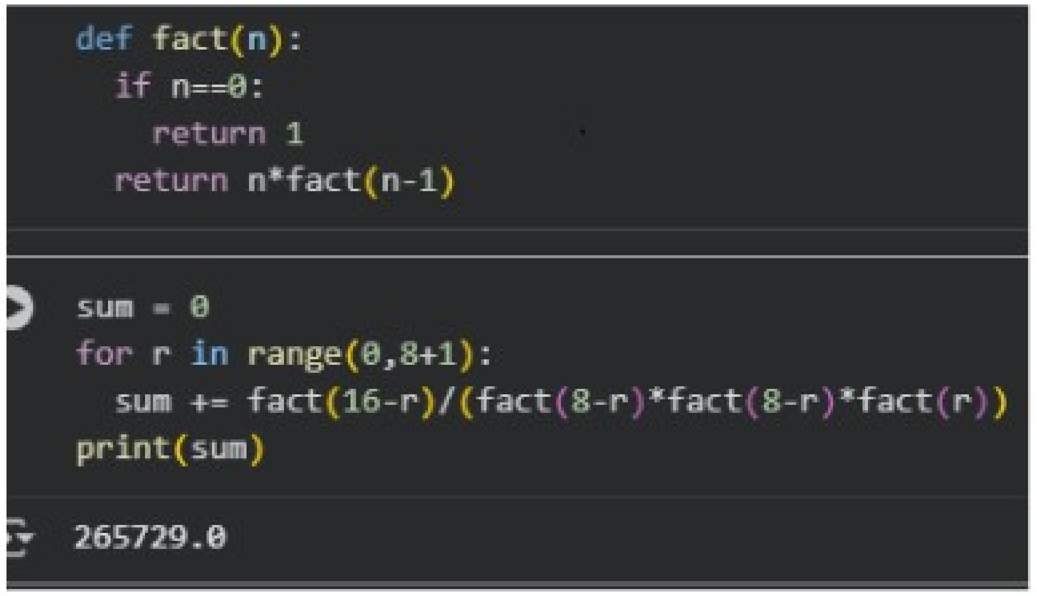
Taking *diagonal moves also* into account the number of paths is
∑ (M+N-R)! / (M-R)! × (N-R)! × R!
where R varies from 0 to min(M,N) = 8
and stands for the number of diagonal moves in the path

N=M=8

$\nearrow$ $\frac{16!}{8!\,8!}$

R: 0 1 2 3 4 5 6 7 8

$\frac{16!}{8!\,8!}$ $\frac{15!}{7!\,7!\,1!}$



```
def fact(n):
  if n==0:
    return 1
  return n*fact(n-1)
```

```
sum = 0
for r in range(0,8+1):
  sum += fact(16-r)/(fact(8-r)*fact(8-r)*fact(r))
print(sum)
```

```
265729.0
```