$S = \{a_1, \dots, a_n\}$ jobs to be scheduled

s_i = start time of job i

f_i = finish

sorted so that $i \le j \Leftrightarrow f_i \le f_j$

S_{ij} = all jobs that start after a_i is done
end before a_j starts

$= \{a_k \in S \mid f_i \le s_k < f_k \le s_j\}$

add fake jobs a_0, a_{n+1}
$f_0 = 0$
$s_{n+1} = \infty$

$S_{ij} = \{\}$ if $i \ge j$

$S_{0,n+1}$ represents the whole problem

S_{ij} (where $i<j$) represents a sub-problem

if the solution to S_{ij} contains a_k
then we will have to solve sub-problems S_{ik} and S_{kj}
solution to S_{ij} is then (solution to S_{ik}) $\cup$ (sol to S_{kj}) $\cup \{a_k\}$

A[i][j] is to hold the number of jobs in the biggest subset of compatible jobs in S_{ij}

if a_k appears in the solution to S_{ij}
then A[i][j] = A[i][k] + 1 + A[k][j]

try all possible values of k

$$A[i][j] = \begin{cases} 0 & \text{if } S_{ij} = \{\} \\ \max\limits_{i<k<j} \left(A[i][k] + 1 + A[k][j] \right) & \text{otherwise} \end{cases}$$

If S_{ij} is any sub-problem
and a_m is the job in S_{ij} with earliest finish time

$$f_m \leq f_x \text{ for all } x \text{ in } i \ldots j$$

then S_{im} is empty
and a_m is in the optimal solution to S_{ij}

$\therefore$ to solve S_{ij}:

take a_i
and then solve $S_{i+1,j}$

i:	0	1	2	3	4	5	6	7	8	9	10	$\overset{n}{\overset{\shortparallel}{11}}$	12
s_i:	-	1	3	0	5	3	5	6	8	8	2	12	∞
f_i:	0	4	5	6	7	8	9	10	11	12	13	14	-

use a_1, left with $\{4, 6, 7, 8, 9, 11\}$
use a_4, left with $\{8, 9, 11\}$
use a_8, left with $\{11\}$
use a_{11}, left with $\{\}$, all done.