Artificial Intelligence

Lab Task

Name: Ahmed Kasteer

Roll Number: 20F-0336

Section: 6C

Code:

import itertools  
  
jobs = {'A': (3, 5), 'B': (5, 6), 'C': (9, 16), 'D': (7, 14)}  
  
def alpha\_beta(jobs, seq, alpha, beta, depth):  
 if len(seq) == len(jobs):  
 delay = sum(max(0, sum(jobs[j][0] for j in seq[:i + 1]) - jobs[seq[i]][1]) for i in range(len(seq)))  
 return (seq, delay)  
  
 remaining\_jobs = set(jobs.keys()) - set(seq)  
  
 if depth == 0 or len(remaining\_jobs) == 0:  
 return (seq, float('inf'))  
  
 best\_sequence, best\_delay = None, float('inf')  
  
 for next\_job\_seq in itertools.permutations(remaining\_jobs):  
 new\_seq = seq + next\_job\_seq  
  
 new\_delay = sum(  
 max(0, sum(jobs[j][0] for j in new\_seq[:i + 1]) - jobs[new\_seq[i]][1]) for i in range(len(new\_seq)))  
  
 if new\_delay < best\_delay:  
 best\_sequence, best\_delay = new\_seq, new\_delay  
  
 if best\_delay <= alpha:  
 break  
  
 alpha = max(alpha, best\_delay)  
  
 return (best\_sequence, best\_delay)  
  
  
sequences\_dict = {}  
for seq in itertools.permutations(jobs.keys()):  
 \_, delay = alpha\_beta(jobs, seq, float('-inf'), float('inf'), len(jobs))  
 sequences\_dict[seq] = delay  
  
for seq, delay in sequences\_dict.items():  
 print ("Order after pruning: ")  
 print(seq, ':', delay)

Output:



