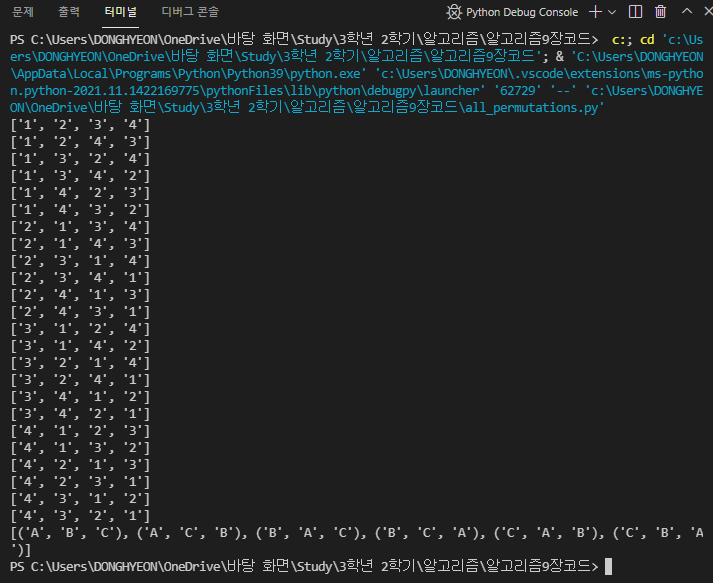
**12주차 과제**

1. **백트래킹을 이용하여 집합 {1,2,3,4}에 대한 모든 가능한 원소의 순서를 나열하세요. (교재 프로그램 사용하여 결과를 캡처하세요)**



**2. 교재의 N-Queen문제에서 N의 변화에 따라 해의 개수가 몇 개인지 알고자 합니다. N을 입력받아 4부터 N까지의 해의 개수를 각각 구하고 이를 출력하는 프로그램을 작성하세요.**

**ex)**

**N을 입력하세요 : 10**

**4 -Queen = 2**

**5 -Queen = 10**

**6 -Queen = 4**

**7 -Queen = 40**

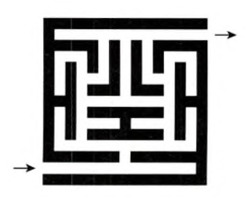
**8 -Queen = 92**

**9 -Queen = 352**

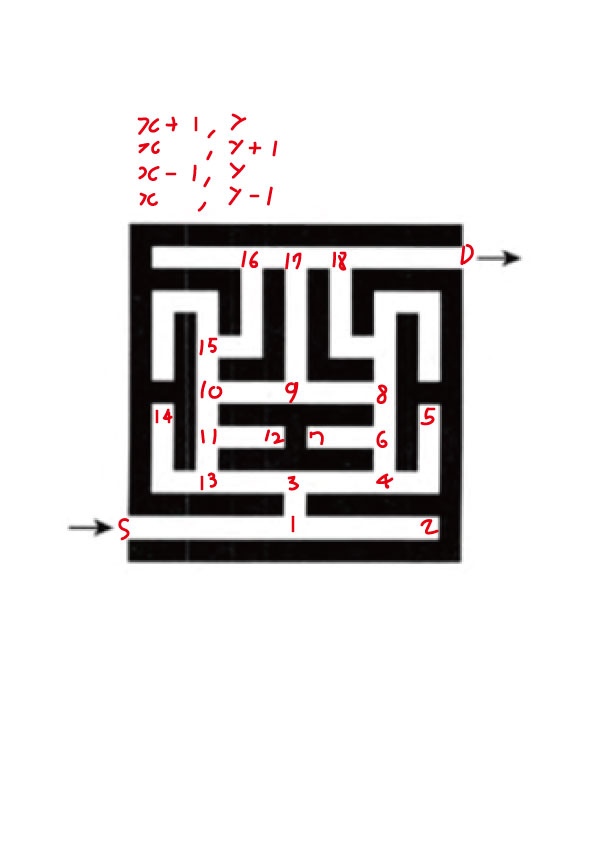
**10 -Queen = 724**

|  |
| --- |
| #소스코드  from typing import Counter  def isSafe(board, x, y):      N = len(board)      for i in range(y):          if board[i][x] == 1: return False   # 세로방향 검사      for i, j in zip(range(y-1, -1, -1), range(x-1, -1, -1)):          if board[i][j] == 1: return False   # \ 방향 검사      for i, j in zip(range(y-1, -1, -1), range(x+1, N)):          if board[i][j] == 1: return False   # / 방향 검사      return True     # 모든 방향으로 OK이면 (x,y)는 safe한 위치    def solve\_N\_Queen(board, y):      N = len(board)      global c      if y == N:                      # 하나의 해 탐색 성공          c += 1          # printBoard(board)           # 화면에 출력          return                      # 백트래킹      for x in range(N):              # 현재 y에서 모든 x를 테스트 함          if isSafe(board, x, y):     # (x,y)에 퀸이 들어갈 수 있으면              board[y][x] = 1         # 넣고              solve\_N\_Queen(board, y+1)# 다음 행 처리: 상태공간트리 탐색              board[y][x] = 0         # 처리가 끝났으니 다시 꺼냄  def printBoard(board):      for i in range(len(board)):          for j in range(len(board)):              if board[i][j] == 1:                  print("Q", end=" ")              else:                  print(".", end=" ")          print()      print()  N = int(input("N을 입력하세요 : "))  for i in range(4, N+1):      c = 0      board = [[0 for i in range(i)] for j in range(i)]      solve\_N\_Queen(board, 0)      print(f"{i} -Queen = {c}") |

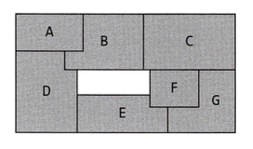
**3. 다음과 같은 미로가 있습니다. 알고리즘 9.4로 이 미로를 탐색하여 출구를 찾는 과정을 보이세요.**

****

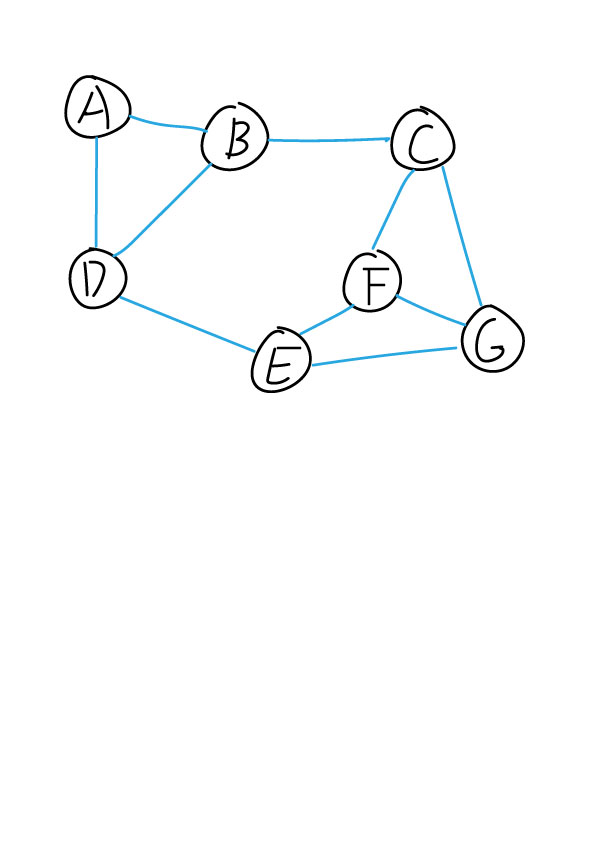
위치 (x, y)가 유효하지 않거나, 더 이상 갈 수 있는 칸이 없는 경우 백트래킹 한다



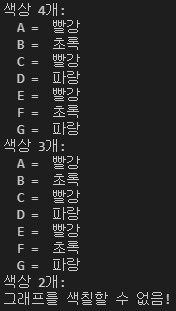
**4. 다음과 같은 지도가 있습니다. 물음에 답하세요.**

****

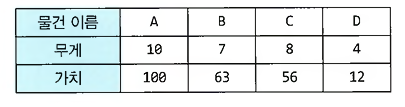
**1) 이 맵을 그래프 형태로 변환하여 그리세요.**



**2) 알고리즘 9.8을 수행하세요. k=2,3,4로 변환해보세요. 최소 색상의 개수는 얼마인가요?**

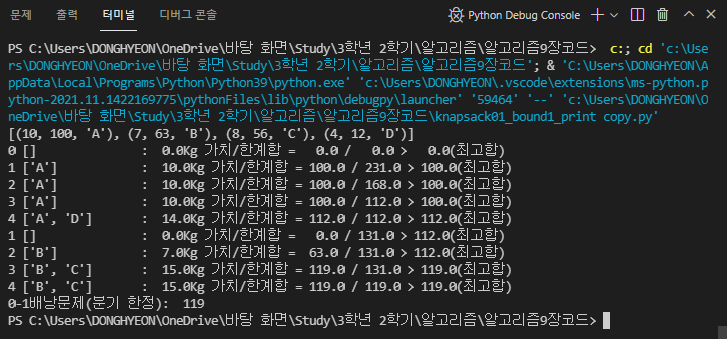


최소 색상의 개수 = 3개

**5. 다음과 같이 주어진 0-1 배낭 채우기 문제를 분기 한정기법으로 해결하세요.**

배낭의 용량이 정의되어 있지 않아 임의로 15로 정했습니다

**1) 한계 가치 계산법으로 bound() (알고리즘 9.10) 사용**



**2) 한계 가치 계산법으로 bound2() (알고리즘 9.11) 사용**

