

파이썬으로 배우는 알고리즘 기초

Chap 6. 분기 한정법



5.6, 3.6

해밀턴 경로와
외판원 문제





5.6 3.6 해밀턴 경로와 외판원 문제



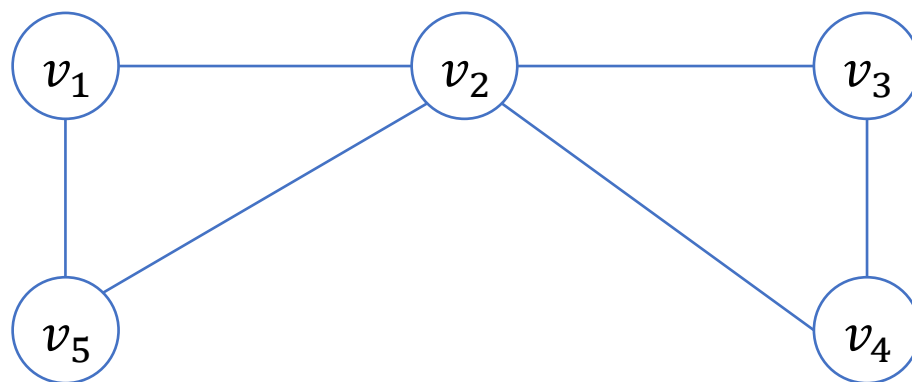
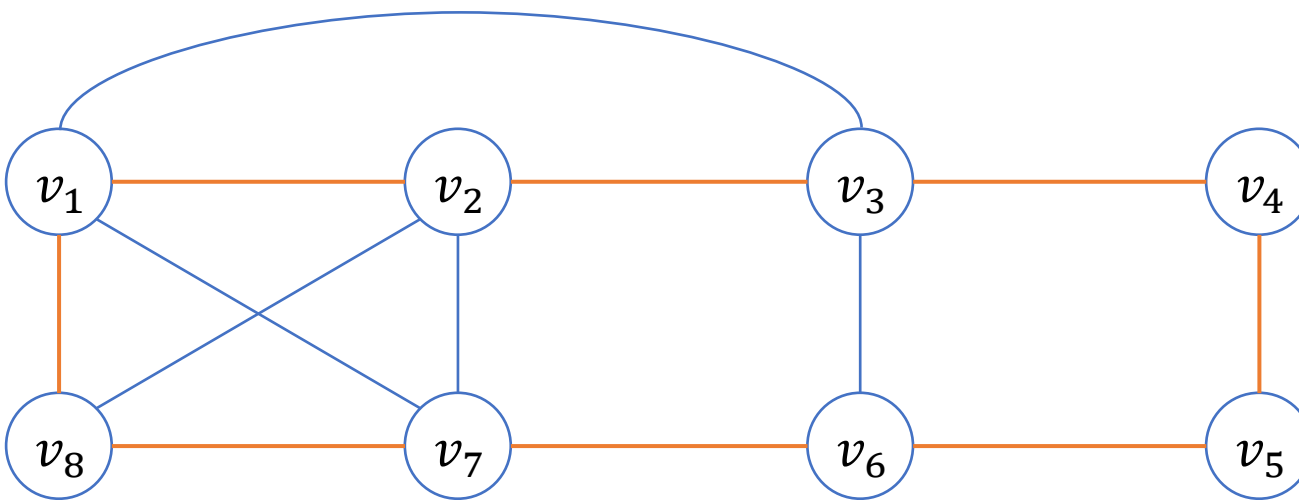
- **해밀턴 경로 문제**: The Hamiltonian Path Problem
 - 주어진 그래프 $G = (V, E)$: 연결된 무방향 그래프
 - 한 정점에서 출발하여 그래프의 모든 정점을 한 번씩만 방문하고
 - 다시 원래 출발한 정점으로 되돌아오는 경로
 - cf) **오일러 경로** (Eulerian Path): 모든 간선을 한 번씩만 방문. 한 붓 그리기 문제.



5.6 3.6 해밀턴 경로와 외판원 문제

주니온TV@Youtube

자세히 보면 유익한 코딩 채널





5.6 3.6 해밀턴 경로와 외판원 문제



■ 해밀턴 경로: 백트래킹으로 풀기

• 상태공간트리의 구축

- 트리의 레벨 0에서 출발 정점을 선정(경로의 0번째 정점)
- 트리의 레벨 1에서 출발 정점을 제외한 각 정점을 출발 정점 다음에 올 첫째 정점으로 고려한다.
- 트리의 레벨 2에서는 앞서서와 같은 정점들을 각각 둘째 정점으로 고려한다.
- 그 이후 수준에서도 같은 방식으로 계속
- 마지막으로, 수준 $n - 1$ 에서 정점들을 $(n - 1)$ 째 정점으로 고려한다.



5.6 3.6 해밀턴 경로와 외판원 문제



Algorithm 5.6: Backtracking for the Hamiltonian Circuits Problem

```
def hamiltonian (i, W, vindex):  
    n = len(W) - 1  
    if (promising(i, W, vindex)):  
        if (i == (n - 1)):  
            print(vindex[0:n])  
        else:  
            for j in range(2, n + 1):  
                vindex[i + 1] = j  
                hamiltonian(i + 1, W, vindex)
```



5.6 3.6 해밀턴 경로와 외판원 문제



- 해밀턴 경로: 백트래킹으로 풀기
 - 인접 행렬 $W[i][j]$:
 - i 번째 정점과 j 번째 정점간에 간선이 있으면 True, 없으면 False
 - 가지치기를 위한 유망 함수의 조건
 1. $(n - 1)$ 째 정점은 출발 정점과 반드시 인접해 있어야 한다.
 2. 경로 상의 i 번째 정점은 그 경로에서 $(i - 1)$ 번째 정점과 반드시 인접해야 한다.
 3. i 번째 정점은 그 앞에 오는 $(i - 1)$ 개의 정점들 중 하나가 될 수 없다.



5.6 3.6 해밀턴 경로와 외판원 문제



Algorithm 5.6: Backtracking for the Hamiltonian Circuits Problem

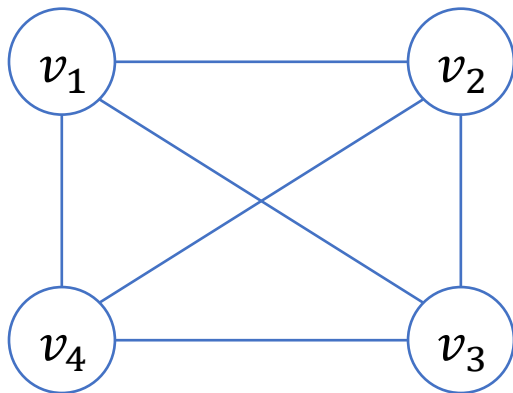
```
def promising (i, W, vindex):  
    flag = True  
    if ((i == (n - 1)) and not W[vindex[n-1]][vindex[0]]):  
        flag = False  
    elif ((i > 0) and not W[vindex[i-1]][vindex[i]]):  
        flag = False  
    else:  
        j = 1  
        while (j < i and flag):  
            if (vindex[i] == vindex[j]):  
                flag = False  
            j += 1  
    return flag
```



5.6 3.6 해밀턴 경로와 외판원 문제

`n = 4`

```
edges = [
    [1, 2],
    [1, 3],
    [1, 4],
    [2, 3],
    [2, 4],
    [3, 4]
]
```



```
W = [[False] * (n + 1) for _ in range(n + 1)]
```

```
for e in edges:
```

```
    W[e[0]][e[1]] = W[e[1]][e[0]] = True
```

```
vindex = [-1] * (n + 1)
```

```
vindex[0] = 1
```

```
hamiltonian(0, W, vindex)
```

[1, 2, 3, 4]

[1, 2, 4, 3]

[1, 3, 2, 4]

[1, 3, 4, 2]

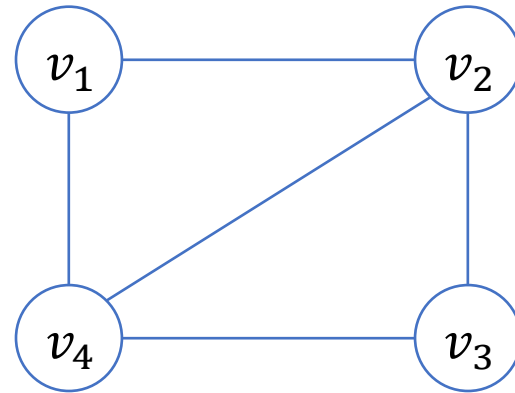
[1, 4, 2, 3]

[1, 4, 3, 2]



5.6 3.6 해밀턴 경로와 외판원 문제

```
n = 4
edges = [
    [1, 2],
    [1, 4],
    [2, 3],
    [2, 4],
    [3, 4]
]
```



[1, 2, 3, 4]

[1, 4, 3, 2]



5.6 3.6 해밀턴 경로와 외판원 문제



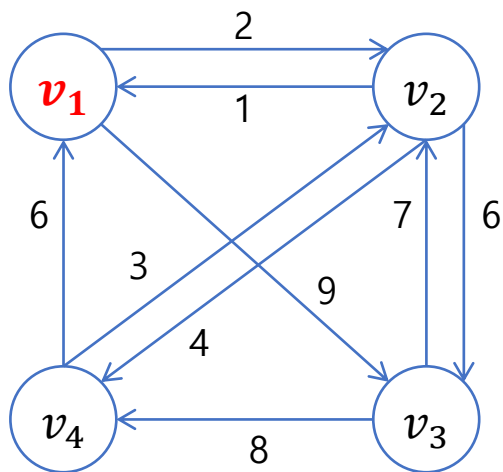
- **외판원 문제**: (TSP: The Traveling Salesperson Problem)
 - 외판원 문제의 이해:
 - 외판원이 20개 도시로 판매 출장을 계획하고 있다고 가정하자.
 - 외판원은 출발 도시에서 모든 도시를 방문하고 다시 되돌아와야 한다.
 - 출장시간(또는 비용)을 최소로 줄이기 위한 방문 계획을 구하고 싶다.
 - 주어진 그래프는 가중치 있는 방향 그래프라고 가정
 - 가중치 있는 그래프에서 해밀턴 경로 문제의 최적화 문제



5.6 3.6 해밀턴 경로와 외판원 문제

■ 외판원 문제의 사례

- 아래와 같은 가중치 있는 방향 그래프에서,
- 가능한 경로의 총 수는 얼마일까?
 - $(n - 1)(n - 2) \cdots 1 = (n - 1)!$
 - 팩토리얼 시간 복잡도: 지수시간 보다 더 나쁘다!



- Starting vertex: v_1
- 3 tours and their path lengths
 - $v_1 v_2 v_3 v_4 v_1 : 22$
 - $v_1 v_3 v_2 v_4 v_1 : 26$
 - $v_1 v_3 v_4 v_2 v_1 : \mathbf{21 \text{ (optimal tour)}}$



주니온TV@Youtube

자세히 보면 유익한 코딩 채널

<https://bit.ly/2JXXGqz>

주니온TV@Youtube

자세히 보면 유익한 코딩 채널

- 여러분의 **구독**과 **좋아요**는 강의제작에 큰 힘이 됩니다.
- 강의자료 및 소스코드: **구글 드라이브**에서 다운로드
(다운로드 주소는 영상 하단 설명란 참고)

<https://bit.ly/3fN0q8t>