Python/알고리즘
파이썬(python) 알고리즘 - 최단 경로 (다익스트라)
2021. 5. 9. 22:36

최단 경로 찾기 (다익스트라)

개요

항목	내용
언제 쓰는가?	**양의 가중치가 있는 방향/양방향 그래프에서 최단경로 찾기** **하나의 노드에서 출발하여 다른 모든 노드로 가는 최소/최대 비용 계산** **(노드+간선)*log(간선) 이 2천만 이하 (python 기준)**
그래프 유형	**방향/양방향, 가중치**
가중치	**양(o) 음(x)**
시간 복잡도	**O((V+E)logV), ((V+E)logV <=20,000,000, n=총 노드 개수)** 각 노드마다 미방문 노드 중 출발점~현재까지의 계산된 최단 거리 노드를 찾는데 O(VlogV) 각 노드마다 이웃한 노드의 최단 거리를 갱신할 때 O(ElogV)
알고리즘 분류	**다이나믹 프로그래밍, 그리디 알고리즘** 매 순간 최저 비용의 노드를 선택하며, 최저 비용은 변하지 않기 때문에 그리디 알고리즘
사용 자료구조	**우선순위 큐 (heapq)**

구현

항목	내용

Algorithm

다음글
파이썬(python) 알고...

파이썬(python) 알고리즘 - 최단 경로 (다익스트라)

자료구조 생성, 초기화

```
1. 인접 리스트 생성
for i in range(e):
    a,b,w = map(int, sys.stdin.readline().split())
    graph[a].append((w,b)) →tuple(비용, 다음노드)

2. 출발 노드 설정
k = 시작 노드 번호

3. 최단거리 테이블 전체를 INF(최대값)로 초기화
distance_arr = [INF] * (v+1)

4. 최단거리 테이블 출발지점을 0으로 초기화
distance_arr[k] = 0

5. 우선순위 큐 생성
heap = []
heapq.heappush(heap,(0,k))
```

메인 로직

```
while heap:
1. 미방문 노드 중 최단 거리가 가장 짧은 것 선택
    dist, cur = heapq.heappop(heap)

2. 방문 여부 검사
    if distance_arr[cur] < dist:
        continue

3. 인접한 모든 노드로 진행 시도
    for n in graph[cur]:
        cost = dist + n[0]
        next = n[1]

4. 다음 노드의 최단 거리 테이블 값이 현재→다음노드로 가는 비용보다 비싸면 갱신, 큐 삽입
        if distance_arr[next] > cost :
            distance_arr[next] = cost
            heapq.heappush(heap,(cost,next))
```

문제 예시 & 코드

1753번: 최단경로

www.acmicpc.net

Algorithm

다음글
파이썬(python) 알고...

```
import sys, heapq

INF = int(1e9)
v,e = map(int, sys.stdin.readline().split())
k = int(sys.stdin.readline())
graph = [[] for _ in range(v+1)]

for i in range(e):
    a,b,w = map(int, sys.stdin.readline().split())
    graph[a].append((w,b))

distance_arr = [INF] * (v+1)
distance_arr[k] = 0

heap = []
heapq.heappush(heap,(0,k))
while heap:
    dist, cur = heapq.heappop(heap)
    if distance_arr[cur] < dist:
        continue
    for n in graph[cur]:
        cost = dist + n[0]
        next = n[1]
        if distance_arr[next] > cost :
            distance_arr[next] = cost
            heapq.heappush(heap,(cost,next))

for i in range(1,v+1):
    if distance_arr[i] == INF:
        print("INF")
    else:
        print(distance_arr[i])
```

응용

최단 경로의 비용 뿐만 아니라 그때의 경로까지 구하기

1. 경로를 저장하는 배열을 생성하고, 자기 자신으로 초기화.

```
trace_arr = [[] for _ in range(n)]

for i in range(n):
    trace_arr[i].append(i)
```

2. 우선순위 큐에 넣을 때마다 경로 추가해주기

다음 노드의 최단경로 = 현재까지의 누적 경로 + 다음노드

```
# 큐에 넣을때
trace_arr[next_pos] = trace_arr[cur_pos] + [next_pos]
```

11779번: 최소비용 구하기 2

www.acmicpc.net

Algorithm

다음글
파이썬(python) 튜플...

파이썬(python) 알고리즘 - 최단 경로 (다익스트라)

```
import sys, heapq
n = int(sys.stdin.readline())
m = int(sys.stdin.readline())
INF = n*100000+1
arr = [[INF] * n for _ in range(n)]
dist_arr = [INF]*n
trace_arr = [[] for _ in range(n)]

for i in range(m):
    s,e,cost = map(int,sys.stdin.readline().split())
    arr[s-1][e-1] = min(arr[s-1][e-1], cost)

for i in range(n):
    trace_arr[i].append(i+1)

s,e = map(int,sys.stdin.readline().split())
s,e = (s-1, e-1)
dist_arr[s] = 0

q = []
heapq.heappush(q, (0, s))
while q:
    cur_dist, cur_pos = heapq.heappop(q)
    if dist_arr[cur_pos] < cur_dist:
        continue
    for i in range(n):
        if arr[cur_pos][i] != INF :
            next_pos, move_dist = (i, arr[cur_pos][i])
            next_dist = cur_dist + move_dist
            if dist_arr[next_pos] > next_dist:
                dist_arr[next_pos] = next_dist
                heapq.heappush(q, (next_dist, next_pos))
                trace_arr[next_pos] = trace_arr[cur_pos] + [next_pos+1]

print(dist_arr[e])
print(len(trace_arr[e]))
for i in trace_arr[e]:
    sys.stdout.write(str(i)+ ' ')
```


'**Python** > **알고리즘**' 카테고리의 다른 글

파이썬(python) 가장 긴 증가하는 부분수열 (LIS) (2)	2021.08.27
파이썬(python) 투 포인터 (0)	2021.05.13
파이썬(python) 알고리즘 - 동적 계획법 푸는 방법 (0)	2021.04.28
파이썬(python) 알고리즘 - DFS, BFS (0)	2021.04.22
파이썬 (python) 알고리즘 - 그리디 알고리즘 (0)	2021.04.22

'Python/알고리즘' 카테고리의 다른 글

파이썬(python) 가장 긴 증가하는 부분수열 (LIS)

파이썬(python) 투 포인터

파이썬(python) 알고리즘 - 동적 계획법 푸는 방법

파이썬(python) 알고리즘 - DFS, BFS

REACT

snowman95

(17~19) Unity/Unreal Engine 게임 프로그래머 (20~21) System Administrator _____ (22~) React 웹 프론트앤드 개발자 _____ 깃헙 : https://github.com/snowman95

Algorithm

다음글

파이썬(python) 투 포...

파이썬(python) 알고리즘 - 최단 경로 (다익스트라)

이름 암호

Algorithm

다음글
파이썬(python) 알...