chapter7. 다익스트라 알고리즘

- Hello Coding 그림으로 개념을 이해하는 알고리즘 -

TABLE OF CONTENTS

1. 너비우선탐색 vs 다익스트라 알고리즘

너비우선 탐색은 가장 적은 수의 구간을 지나는 가장 짧은 길 즉, 최단경로를 의미합니다.

다익스트라 알고리즘은 가장 빠른 길 즉, 최단 시간 경로를 구할 때 사용합니다.

균일 그래프에서 최단 경로를 계산할 때는 너비우선탐색을 사용합니다.

가중 그래프에서 최단 경로를 계산할 때는 다익스트라 알고리즘을 사용합니다.

가중치가 양수일 때만 다익스트라 알고리즘을 사용할 수 있습니다.

2. 다익스트라 알고리즘 구현원리

다익스트라 알고리즘은 다음 4개의 단계로 이루어집니다.

1. 가장 가격(가중치)이 싼 정점, 즉 도달하는 데 시간이 가장 적게 걸리는 정점을 찾습니다.

2. 이 정점의 이웃 정점에 대해 현재의 가격보다 더 싼 경로가 존재하는지 확인합니다. 만약 존재한다면 가격을 수정합니다.

3. 그래프 상의 모든 정점에 대해 이러한 일을 반복합니다.

4. 최종 경로를 계산합니다.

3. 힙 자료구조

다익스트라 알고리즘을 위해서 힙 자료구조를 알아야 합니다.

이전에 스택 자료구조, 큐 자료구조를 배웠습니다. 힙 자료구조는 뭘까요?

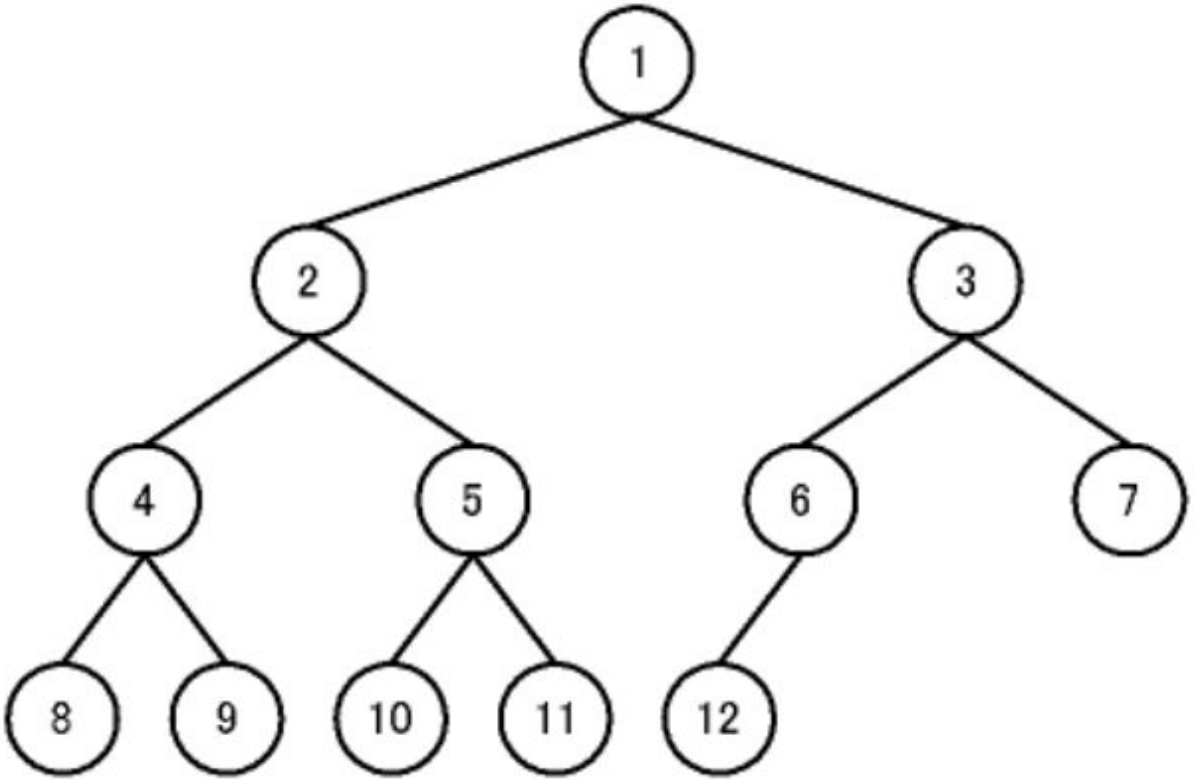
힙 자료구조는 최소값과 최대값을 빠르게 구하려고 만들어낸 자료구조입니다.

힙을 이해하려면 완전이진트리를 알아야 합니다.(모르면 구글링)

루트(완전이진트리 가장 최상단 노드)가 최대값이면 Max heap ,최소값이면 min heap

힙 자료구조는 부모가 자식보다 값이 작음. (Min heap) → 그래서 최상단 노드가 최소값

완전이진트리



자식 두명이어야함.

왼쪽부터 차례대로
채워줘야 함.

마지막 레벨은 자식 숫자
상관없음

heap Tree에 2,6,9,4,7,1순서로 데이터 집어넣을 때 처리과정

```
from heapq import *

a = [10,11,9,2,8,3,7,4,5,6]
heapify(a)
print(heappop(a))
print(heappop(a))
print(heappop(a))
print(heappop(a))
print(a)
```

```
2
3
4
5
[6, 8, 7, 10, 11, 9]
```

일단 힙도 리스트와 생김새는 비슷함.

a 는 리스트이고 안에 숫자 아무렇게 배열함.

여기서 최소값만 끄집어내고 싶음.

heapify함수 사용하면 리스트도 힙으로 바꿔줌.

즉 a는 힙 으로 변함. (원소 순서가 바뀜 맨 왼쪽은 최소값)

heappop()를 사용하면 최소값만 끄집어 낼 수 있으며 나머지 원소들은 힙 자료구조 상태를 그대로 유지.

```python
from heapq import *

heap = []
heappush(heap, 2)
heappush(heap, 6)
heappush(heap, 9)
heappush(heap, 4)
heappush(heap, 7)
heappush(heap, 1)
print(heap)
```

```
[1, 4, 2, 6, 7, 9]
```

〈heappush 동작순서〉

1. 값을 하나씩 트리의 노드에 좌에서 우로 할당.
2. 부모가 자식보다 크면 둘의 순서 뒤바꿈.
3. **루트노드까지 비교를 반복.**