

Python/알고리즘

BFS 완벽 구현하기 - 파이썬

BK_Paul 2021. 2. 1. 13:28

[소셜개발] 커뮤니티 서비스 백엔드 개발

더보기

인사담당자 (HR Manager)

더보기

백엔드 개발자

더보기

커머스 백엔드 개발자 (Node.js)

더보기

나다운 일의 시...
원티드

1. BFS 기본 개념

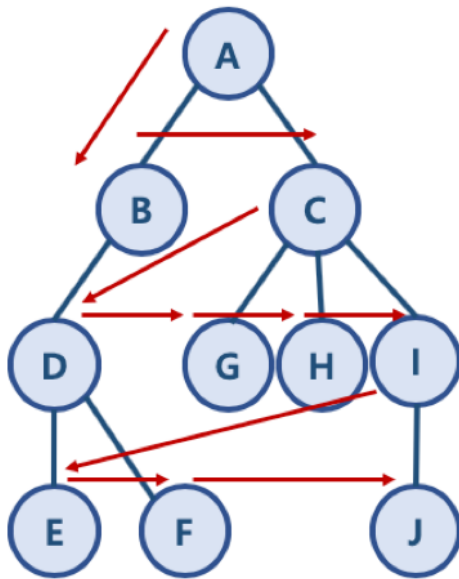
2. BFS 구현 원리

3. BFS 구현 코드

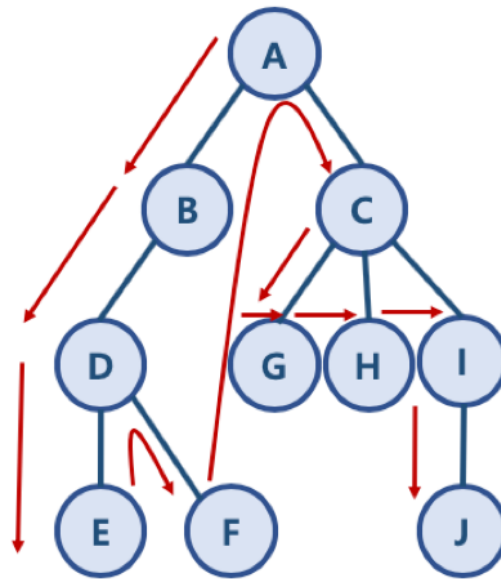
1. BFS 기본 개념

BFS란 **그래프 자료 구조에서 원하는 자료를 찾는 탐색 알고리즘** 중에 하나입니다.

자료를 찾을 때 "수직" 방향으로 자료를 검색할 수도 있고, "**수평**" **방향으로 자료를 검색**할 수 있는데, BFS는 이름에서 추론할 수 있듯이, "수평방향"으로 원하는 노드를 탐색하는 알고리즘입니다.



Breadth First Search



Depth First Search

(DFS는 이 포스팅을 참고해주시기 바랍니다.)

2. BFS 구현 원리

BFS를 구현하기 위해서는 항상 "방문하고자 하는 노드"와 "방문했던 노드"를 나누어서 알고리즘을 구성하는 것이 핵심 원리입니다.

논리를 3단계로 요약하면 다음과 같습니다.

1단계. 시작 노드를 방문했던 노드에 삽입한다.

2단계. 방문할 노드에 시작노드의 Child Node를 삽입한다.

3단계. Child노드를 중심으로 다시 1~2단계를 거쳐 탐색한다.

3. BFS 구현 코드

파이썬에서는 간편하게 리스트 자료 구조를 활용하여 BFS를 구현할 수 있습니다.

(1) 기반 데이터 - 그냥 복붙하세요

```
1 graph = dict()
2
3 graph['A'] = ['B', 'C']
```

CS

```

4 graph['B'] = ['A', 'D']
5 graph['C'] = ['A', 'G', 'H', 'I']
6 graph['D'] = ['B', 'E', 'F']
7 graph['E'] = ['D']
8 graph['F'] = ['D']
9 graph['G'] = ['C']
10 graph['H'] = ['C']
11 graph['I'] = ['C', 'J']
12 graph['J'] = ['I']

```

(2) 구현 코드

```

1 def bfs(graph, start_node):
2     need_visited, visited = [], []
3     need_visited.append(start_node)
4
5
6     while need_visited:
7         node = need_visited[0]
8         del need_visited[0]
9
10        if node not in visited:
11            visited.append(node)
12            need_visited.extend(graph[node])
13    return visited

```

CS

BFS와 DFS의 가장 큰 차이점은 **While문 다음에 어떤 자료를 우선적으로 추출**하느냐입니다.

DFS 같은 경우 리스트의 가장 끝에 있는 데이터를 기준으로 추출하지만,

BFS는 리스트의 가장 처음에 있는 인자를 받습니다.

그래서 need_visited[0]이라는 코드를 작성한 것입니다.

```

[12]: def bfs(graph, start_node):
      need_visited, visited = [], []
      need_visited.append(start_node)

      while need_visited:
          node = need_visited[0]
          del need_visited[0]

          if node not in visited:
              visited.append(node)
              need_visited.extend(graph[node])
      return visited

[13]: bfs(graph, 'A')

[13]: ['A', 'B', 'C', 'D', 'G', 'H', 'I', 'E', 'F', 'J']

```

BFS 구현 예시

2

구독하기

'Python > 알고리즘' 카테고리의 다른 글

[Python]버블정렬 알고리즘 코드 및 예시로 마스터 하기 (0)	2022.08.23
DFS 완벽 구현하기 [Python] (4)	2021.01.29
그래프의 이해 - 고도 알고리즘을 위한 기초 개념 (0)	2021.01.17
탐색 알고리즘 - 이진 탐색 (0)	2021.01.11
탐색 알고리즘 - 순차 탐색(Sequential Search) (0)	2021.01.09

태그 #BFS, #BFS구현, #Python, #그래프구조, #데이터구조, #알고리즘, #파이썬

'Python/알고리즘' Related Articles

```
"""heap"""
def heap_sort(data):
    n = len(data)
    for i in range(n // 2 - 1, -1, -1):
        _sift_down(data, i)
    for i in range(n - 1, 0, -1):
        data[i], data[0] = data[0], data[i]
        _sift_down(data, 0)
    return data

def _sift_down(data, i):
    n = len(data)
    while True:
        left = 2 * i + 1
        right = 2 * i + 2
        if left > n - 1:
            break
        if right > n - 1:
            if data[i] > data[left]:
                data[i], data[left] = data[left], data[i]
            break
        if data[i] > data[left] and data[i] > data[right]:
            data[i], data[left] = data[left], data[i]
            i = left
        elif data[i] > data[right]:
            data[i], data[right] = data[right], data[i]
            i = right
        else:
            break
```

```
def dfs_recursive(graph, start, visited = []):
    ed.append(start)
    for node in graph[start]:
        if node not in visited:
            dfs_recursive(graph, node, visited)
    n_visited
```

```
def bfs(graph, 'A')
    queue = ['A']
    while queue:
        node = queue.pop(0)
        for neighbor in graph[node]:
            if neighbor not in visited:
                queue.append(neighbor)
                visited.append(neighbor)
```

```
def h_search(data, search):
    if len(data) == 1 and data[0] == search:
        return "찾았음"
    if len(data) == 1 and data[0] != search:
        return "못 찾음"
    if len(data) == 0:
        return "찾을 게도 없어"
    mid = len(data) // 2
    if search == data[mid]:
        return "찾았음"
    else:
        if search < data[mid]:
            return h_search(data[:mid], search)
        else:
            return h_search(data[mid:], search)
```

- [Python]버블정렬 알고리즘 코드 및 예시로 마스터 하기
- DFS 완벽 구현하기 [Python]
- 그래프의 이해 - 고도 알고리즘을 위한 기초 개념
- 탐색 알고리즘 - 이진 탐색

이름

암호

☐ Secret

여러분의 소중한 댓글을 입력해주세요.

댓글달기

데이터로 하는 마... 구독하기

