컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 35강 - 너비 우선 탐색

Fast campus

학습 목표

너비 우선 탐색

1) 너비 우선 탐색의 개념을 이해합니다.

2) 너비 우선 탐색을 C언어를 이용하여 구현할 수 있습니다.

너비 우선 탐색

너비 우선 탐색

너비 우선 탐색(Breadth First Search)은 너비를 우선으로 하여 탐색을 수행하는 탐색 알고리즘입니다. DFS와 마찬가지로 맹목적으로 전체 노드를 탐색하고자 할 때 자주 사용되며 큐(Queue) 자료구조에 기초합니다.

너비 우선 탐색은 고급 그래프 탐색 알고리즘에서 자주 활용되므로 고급 개발자가 되기 위해서는 너비 우선 탐색에 대해 숙지해야 합니다.

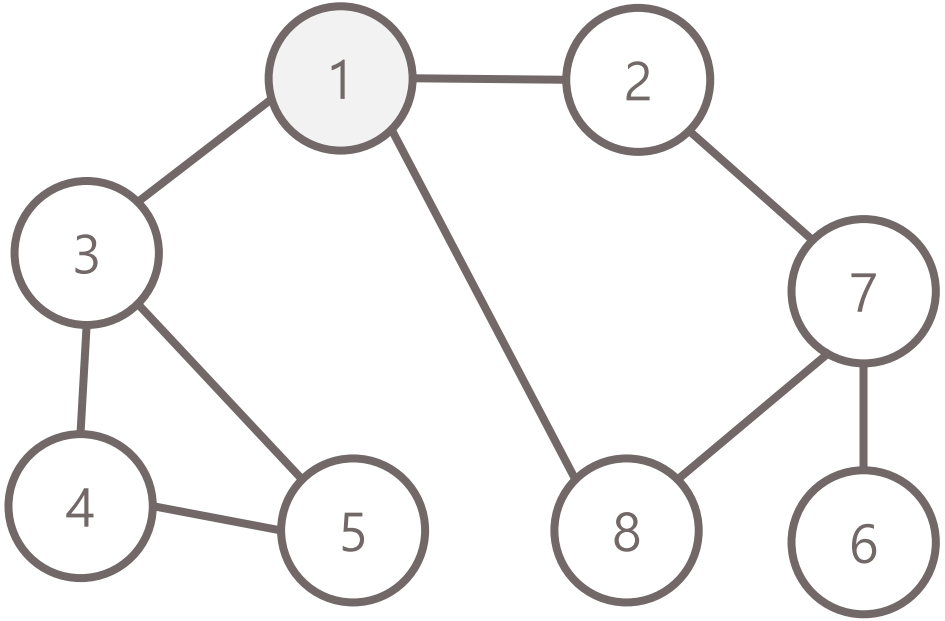
너비 우선 탐색

너비 우선 탐색

1. 탐색 시작 노드를 큐에 삽입하고 방문 처리를 합니다.
2. 큐에서 노드를 꺼내 해당 노드의 인접 노드 중에서 방문하지 않은 노드들을 모두 큐에 삽입하고, 방문 처리를 합니다.
3. 2번의 과정을 더 이상 수행할 수 없을 때까지 반복합니다.

너비 우선 탐색

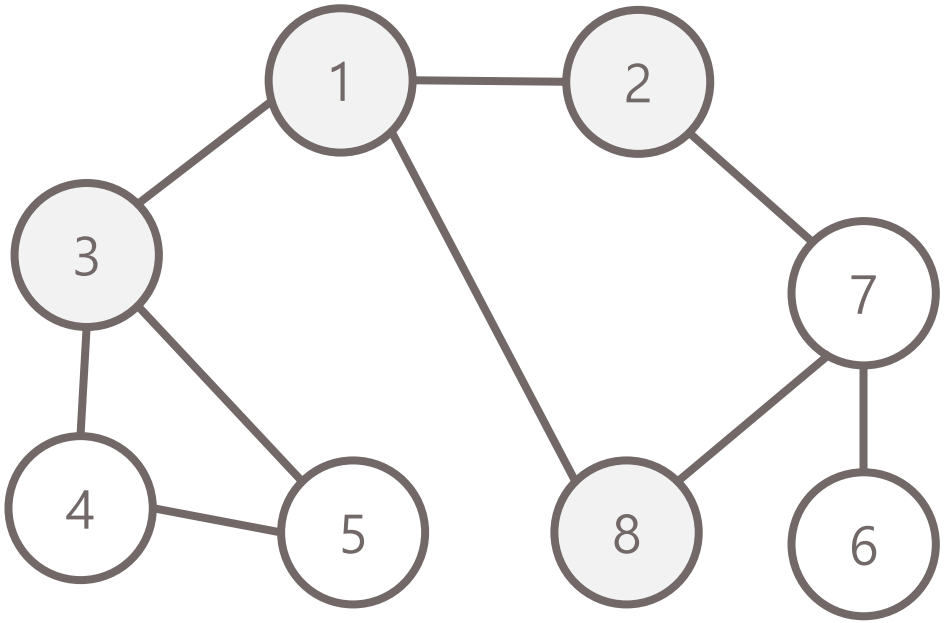
너비 우선 탐색



너비 우선 탐색

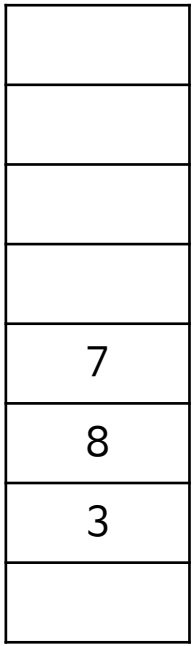
너비 우선 탐색

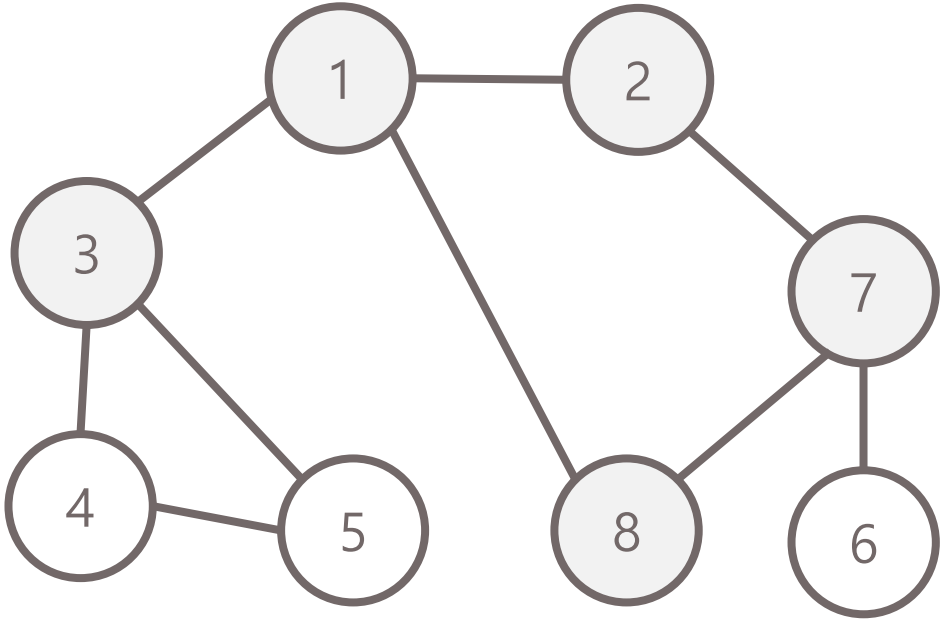
8
3
2



너비 우선 탐색

너비 우선 탐색

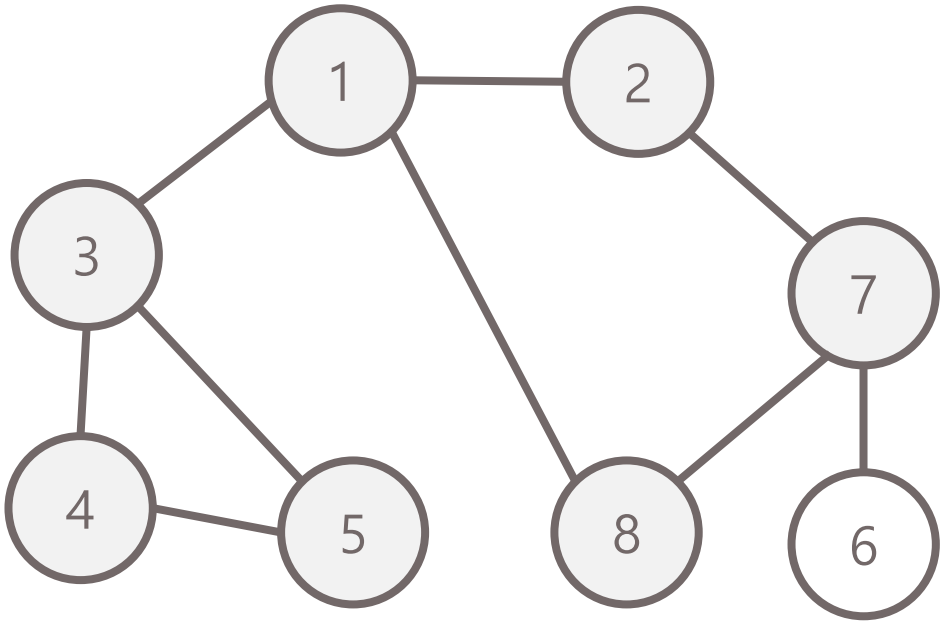




너비 우선 탐색

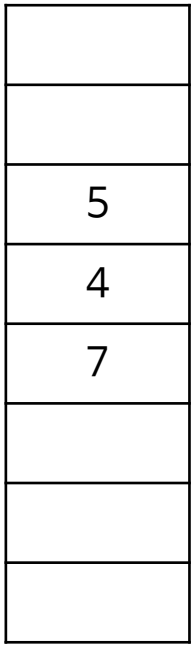
너비 우선 탐색

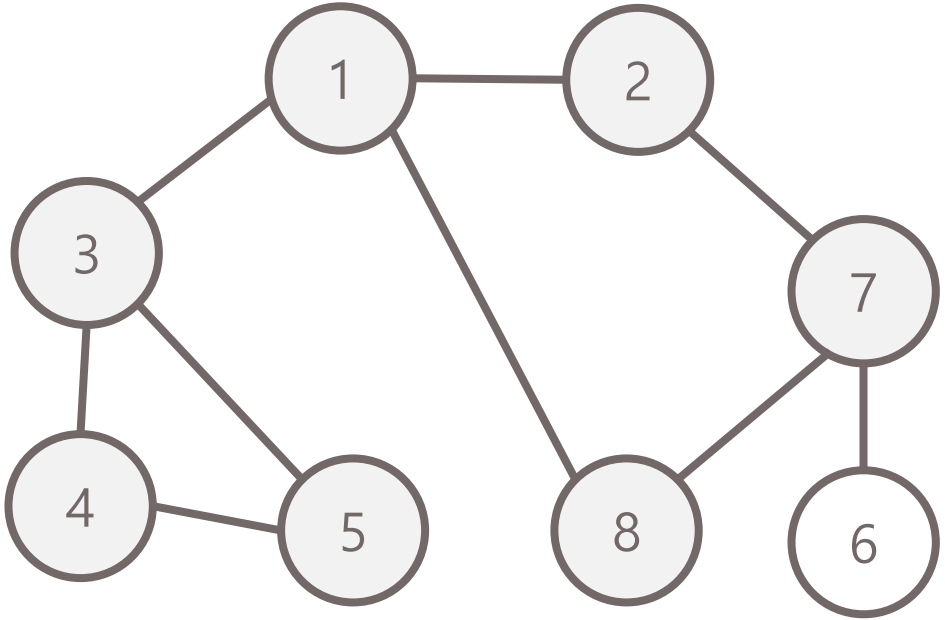
5
4
7
8



너비 우선 탐색

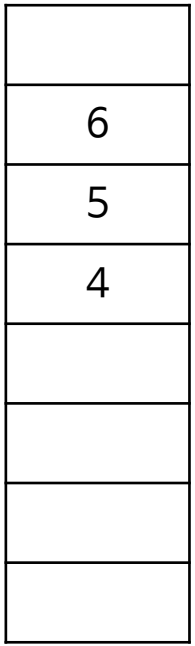
너비 우선 탐색

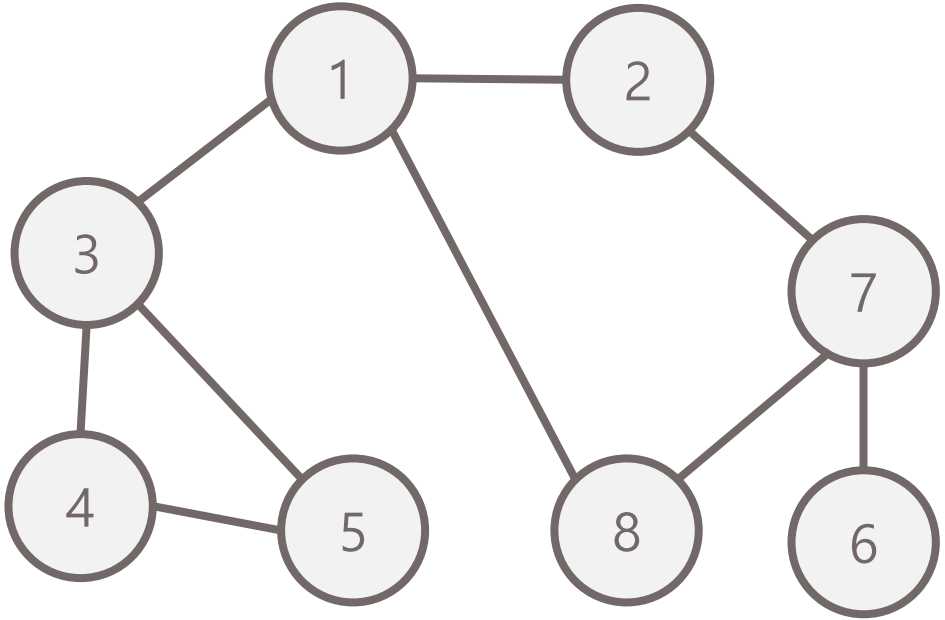




너비 우선 탐색

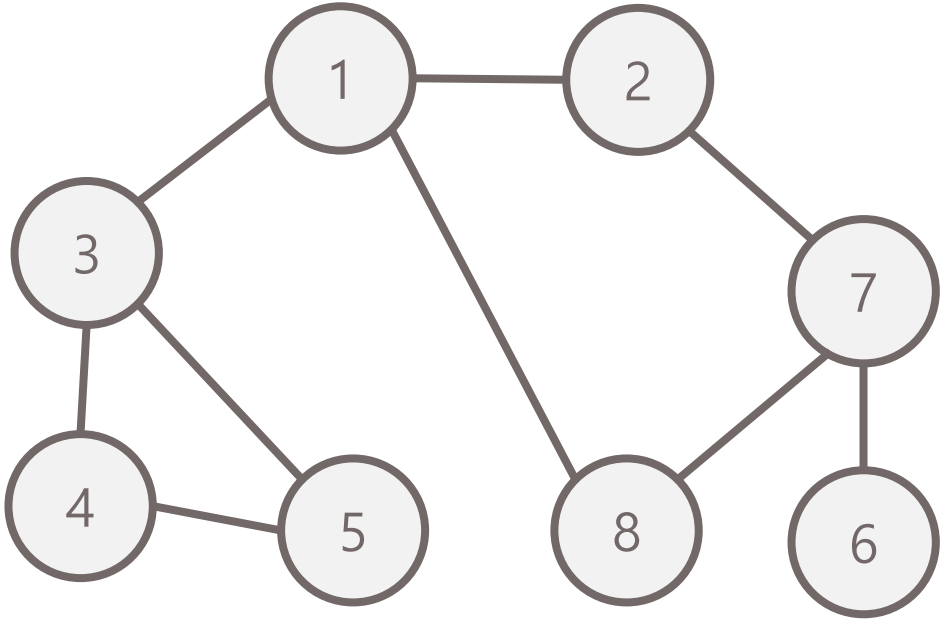
너비 우선 탐색





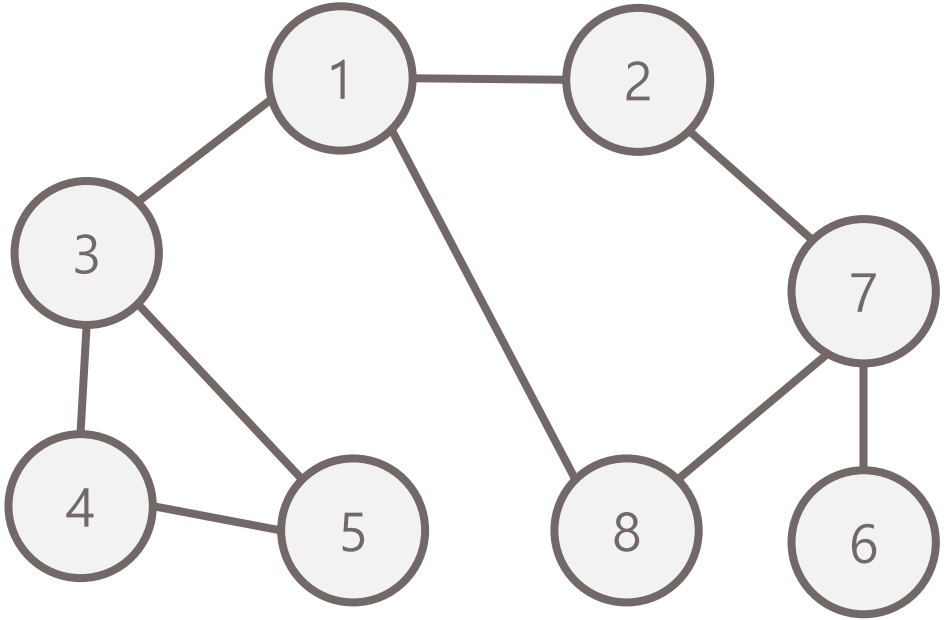
너비 우선 탐색

너비 우선 탐색



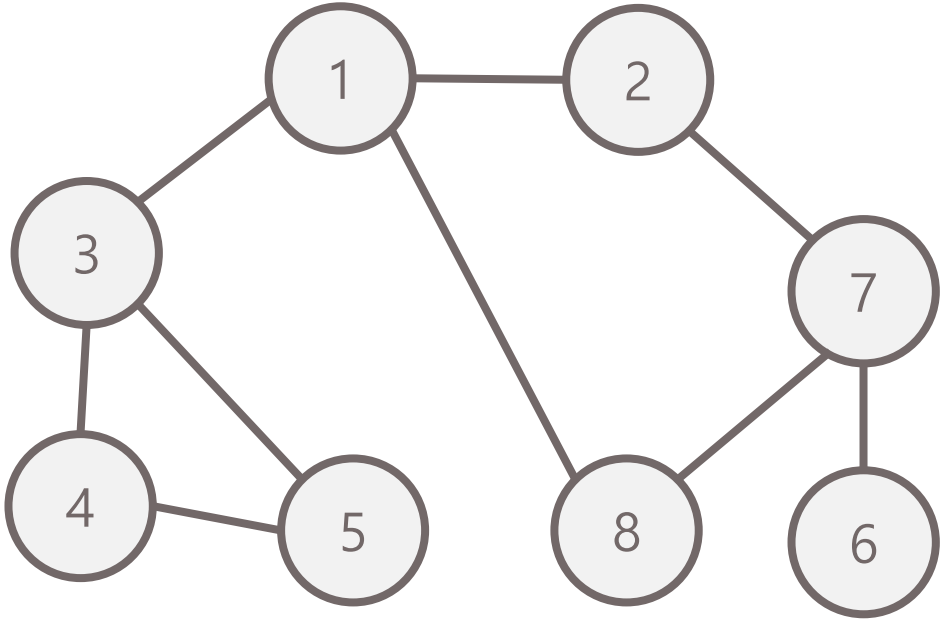
너비 우선 탐색

너비 우선 탐색



너비 우선 탐색

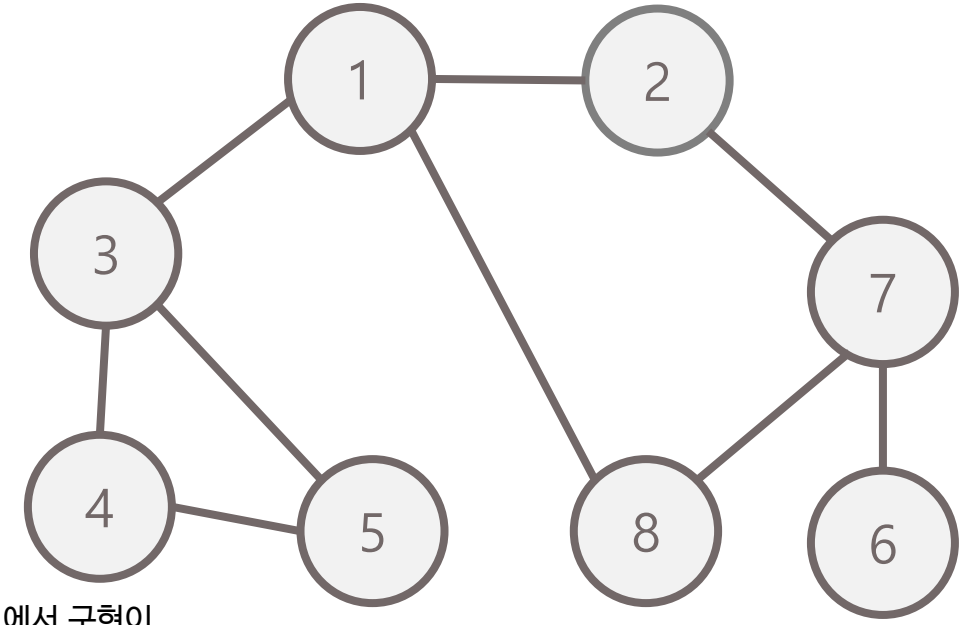
너비 우선 탐색



너비 우선 탐색

너비 우선 탐색

방문 순서: 1 → 2 → 3 → 8 → 7 → 4 → 5 → 6



너비 우선 탐색 알고리즘은 큐(Queue) 자료구조에 기초한다는 점에서 구현이 간단합니다. 실제로 구현함에 있어 큐 STL을 사용하면 좋으며 탐색을 수행함에 있어서 $O(N)$의 시간이 소요됩니다. 일반적인 경우 실제 수행 시간은 DFS보다 좋은 편입니다.

너비 우선 탐색

너비 우선 탐색 1) 연결 리스트 정의

```c
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#define INF 99999999
#define MAX_SIZE 1001

typedef struct {
  int index;
  struct Node *next;
} Node;

typedef struct {
  Node *front;
  Node *rear;
  int count;
} Queue;

Node** a;
int n, m, c[MAX_SIZE];
```

너비 우선 탐색

너비 우선 탐색 2) 연결 리스트 삽입 함수

```
void addFront(Node *root, int index) {
  Node *node = (Node*)malloc(sizeof(Node));
  node->index = index;
  node->next = root->next;
  root->next = node;
}
```

너비 우선 탐색

너비 우선 탐색 3) 큐 삽입 함수

```
void queuePush(Queue *queue, int index) {
  Node *node = (Node*)malloc(sizeof(Node));
  node->index = index;
  node->next = NULL;
  if (queue->count == 0) {
    queue->front = node;
  }
  else {
    queue->rear->next = node;
  }
  queue->rear = node;
  queue->count++;
}
```

너비 우선 탐색

너비 우선 탐색 4) 큐 추출 함수

```
int queuePop(Queue *queue) {
  if (queue->count == 0) {
    printf("큐 언더플로우가 발생했습니다.\n");
    return -INF;
  }
  Node *node = queue->front;
  int index = node->index;
  queue->front = node->next;
  free(node);
  queue->count--;
  return index;
}
```

Fast campus

너비 우선 탐색

너비 우선 탐색 5) 너비 우선 탐색 함수

```
void bfs(int start) {
  Queue q;
  q.front = q.rear = NULL;
  q.count = 0;
  queuePush(&q, start);
  c[start] = 1;
  while (q.count != 0) {
    int x = queuePop(&q);
    printf("%d ", x);
    Node *cur = a[x]->next;
    while (cur != NULL) {
      int next = cur->index;
      if (!c[next]) {
        queuePush(&q, next);
        c[next] = 1;
      }
      cur = cur->next;
    }
  }
}
```

너비 우선 탐색

너비 우선 탐색 6) 너비 우선 탐색 이용해보기

```
int main(void) {
  scanf("%d %d", &n, &m);
  a = (Node**)malloc(sizeof(Node*) * (MAX_SIZE));
  for (int i = 1; i <= n; i++) {
    a[i] = (Node*)malloc(sizeof(Node));
    a[i]->next = NULL;
  }
  for (int i = 0; i < m; i++) {
    int x, y;
    scanf("%d %d", &x, &y);
    addFront(a[x], y);
    addFront(a[y], x);
  }
  bfs(1);
  system("pause");
  return 0;
}
```

너비 우선 탐색

너비 우선 탐색

1) 너비 우선 탐색은 $O(N)$의 시간이 소요되는 전수 탐색 알고리즘입니다.