

# 실행 흐름 모델 & 알고리즘 사고 프레임 정리

이 문서는 코딩 테스트와 문제 해결에서 자주 사용되는 **실행 흐름 모델(Execution Flow Models)** 과 각 모델이 언제 사용되는지, 어떻게 판단하는지를 체계적으로 정리한 가이드입니다.

## 1. 실행 흐름 모델이란?

실행 흐름 모델은

프로그램이 어떤 순서와 전략으로 문제를 처리하는가를 설명하는 사고 구조입니다.

코드 이전 단계의 **생각의 틀**이며, 알고리즘 문제를 보면 가장 먼저 떠올려야 하는 개념입니다.

## 2. 큰 분류 한눈에 보기

분류	핵심 키워드	대표 예시
순차	위에서 아래	기본 코드
분기	if / else	조건 문제
반복	for / while	배열 순회
재귀	자기호출	DFS
스택	LIFO	DFS, 괄호 검사
큐	FIFO	BFS, 최단거리
우선순위 큐	작은 것부터	다익스트라
분할정복	쪼개기	병합정렬
동적계획법	저장/재사용	피보나치
탐욕	매 순간 최선	거스름돈
백트래킹	해보고 되돌림	N-Queen
이벤트	이벤트 발생 시	GUI

## 3. 순차 실행 (Sequential)

### 개념

위에서 아래로 한 줄씩 실행

```
a = 1
b = 2
c = a + b
```

#### 사용 상황

- 단순 계산
  - 입력 → 처리 → 출력 구조
- 

### 4. 분기 실행 (Branching)

```
if x > 0:
    ...
else:
    ...
```

#### 사용 상황

- 조건에 따라 다른 처리
- 

### 5. 반복 실행 (Iteration)

```
for i in range(n):
    ...
```

#### 사용 상황

- 배열 순회
  - 누적합
- 

### 6. 재귀 실행 (Recursion)

```
def f():
    if 종료조건:
        return
    f()
```

#### 특징

- 문제를 작은 문제로 쪼갬
  - DFS, 분할정복의 기반
-

## 7. 스택 기반 모델 (DFS)

### 특징

- 깊게 내려감
- LIFO

```
def dfs(v):  
    visited[v] = True  
    for nxt in graph[v]:  
        if not visited[nxt]:  
            dfs(nxt)
```

### 사용 상황

- 모든 경우의 수
  - 경로 탐색
- 

## 8. 큐 기반 모델 (BFS)

```
from collections import deque  
  
def bfs(start):  
    q = deque([start])  
    visited[start] = True  
    while q:  
        cur = q.popleft()  
        for nxt in graph[cur]:  
            if not visited[nxt]:  
                visited[nxt] = True  
                q.append(nxt)
```

### 특징

- 가까운 것부터
  - 최단거리
- 

## 9. 우선순위 큐 모델

```
import heapq  
heap = []  
heapq.heappush(heap, (0, start))
```

## 사용 상황

- 가중치 있는 최단거리
- 

## 10. 분할 정복 (Divide & Conquer)

### 개념

쪼개기 → 각각 해결 → 합치기

예: 병합정렬, 퀵정렬

---

## 11. 동적 계획법 (DP)

### 개념

- 작은 문제 결과 저장
- 재사용

$$dp[i] = dp[i-1] + dp[i-2]$$

---

## 12. 탐욕법 (Greedy)

### 개념

- 지금 당장 최선 선택

예: 동전 문제

---

## 13. 백트래킹

### 개념

- DFS + 되돌리기

```
def backtrack():
    if 종료:
        return
    for 선택 in 후보:
        선택
        backtrack()
        되돌리기
```

---

## 14. 이벤트 기반 모델

이벤트 발생 시 함수 실행

예: 버튼 클릭

---

## 15. 문제 보고 모델 고르는 법

문제 특징	선택 모델
최단거리	BFS
모든 경우	DFS
최적값 반복	DP
매 순간 선택	Greedy
조합/순열	백트래킹

---

## 16. 핵심 요약

- 알고리즘 = 코드 이전에 **모델 선택**
  - 모델이 보이면 절반 해결
- 

## 17. 추천 학습 순서

1. DFS/BFS
  2. 백트래킹
  3. DP
  4. 그리디
  5. 우선순위 큐
- 

끝.