



# Software Engineering CFG

Team 4

200811441 윤동민

200811458 조원진

200811416 김영훈

200811452 이희봉

# Control Flow Graph

- Control Flow Graph는 컴퓨터 프로그램이 가질 수 있는 다양한 코드의 path들을 시각적으로 표현.
- CFG는 시작 노드, 종료 노드, Basic Block Node, Not Basic Block Node 그리고 decision으로 구성. 시작노드와 종료 노드는 비어있다.
- Node는 프로그래밍 코드의 라인들을 나타내고 있는데 분기문을 포함하지 않은 하나 이상의 순차적이고 연속된 (Basic Block)Node와 if, switch, while문 등 조건문(분기문)을 가지고 있는 (Not basic block)Node가 있다.
- Decision은 실행흐름, 조건부흐름 등을 나타낸다.

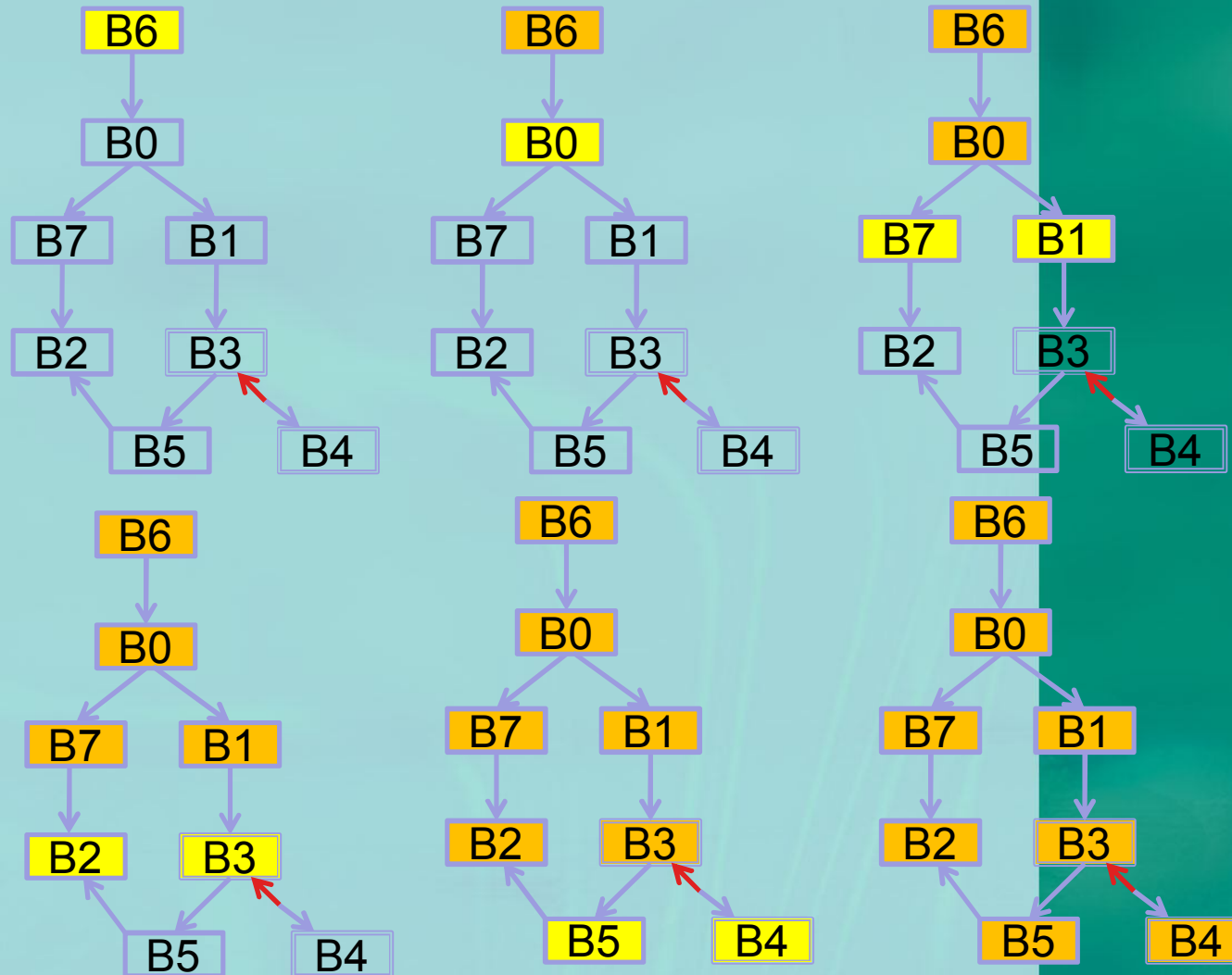
# Control Flow Graph(Cont.)

- BFS, Loop, Hierarchical
- CFG를 생성하는 주요한 목적 중 한 가지는 컴퓨터 프로그램 중 불필요한 부분을 발견하기 위함.
- CFG 생성의 또 다른 목적은 무한루프와 같은 문제들을 해결하는데 도움을 준다.

# 1. BFS Positioning Algorithm

- 이 알고리즘은 그래프에 있는 모든 Node 들을 모두 다 방문 한다. BFS의 동작 원리는 root Node부터 edge를 따라 Child Node를 전부 방문하고 Back 다시 그 Child Node를 방문하는 방식이다.
- BFS 알고리즘의 이점은 height 가 다른 두 개의 알고리즘( Loop, Hierarchical)에 비해 일반적으로 작다는 점이다.

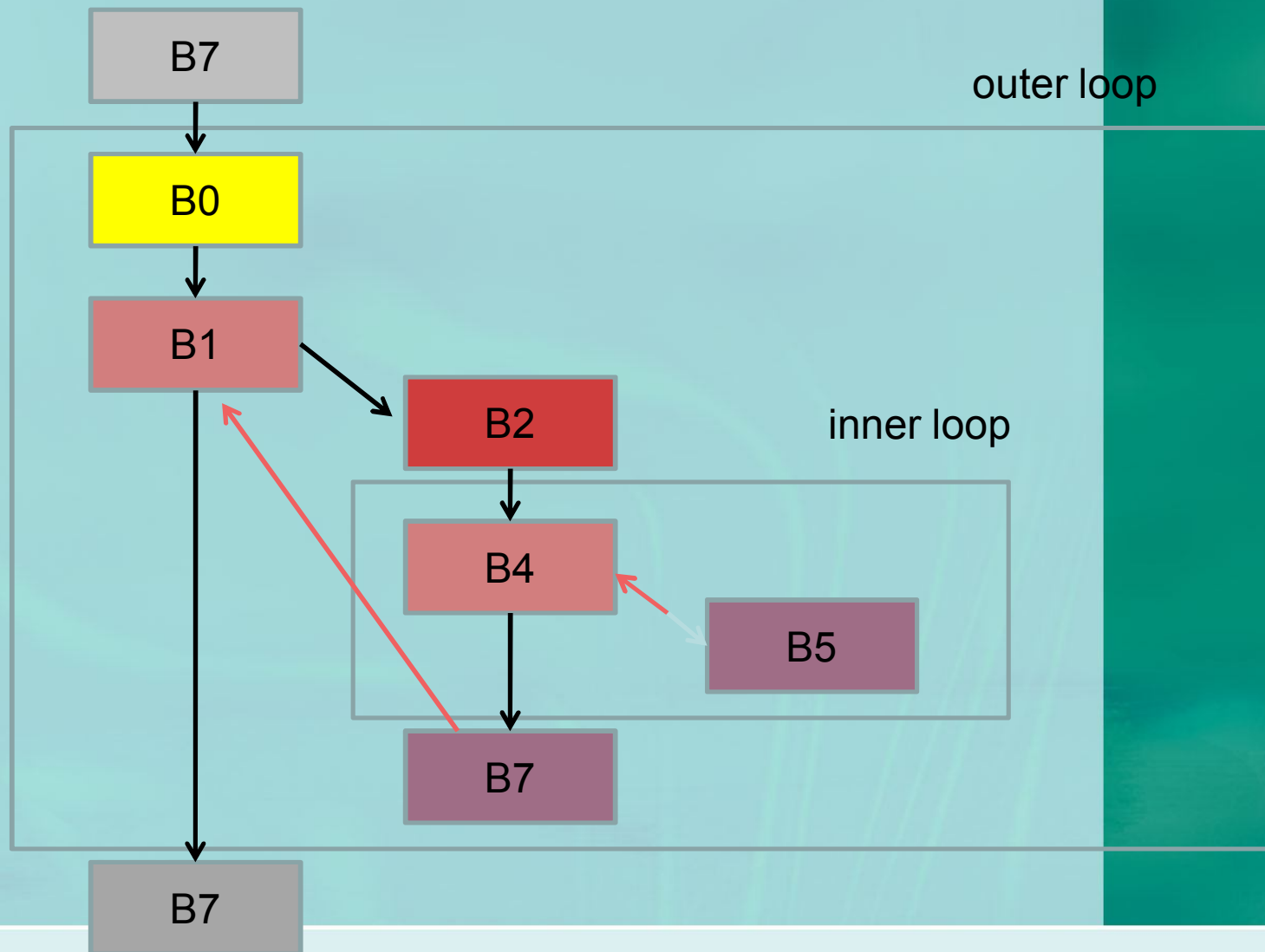
# 1. BFS Positioning Algorithm(Cont.)



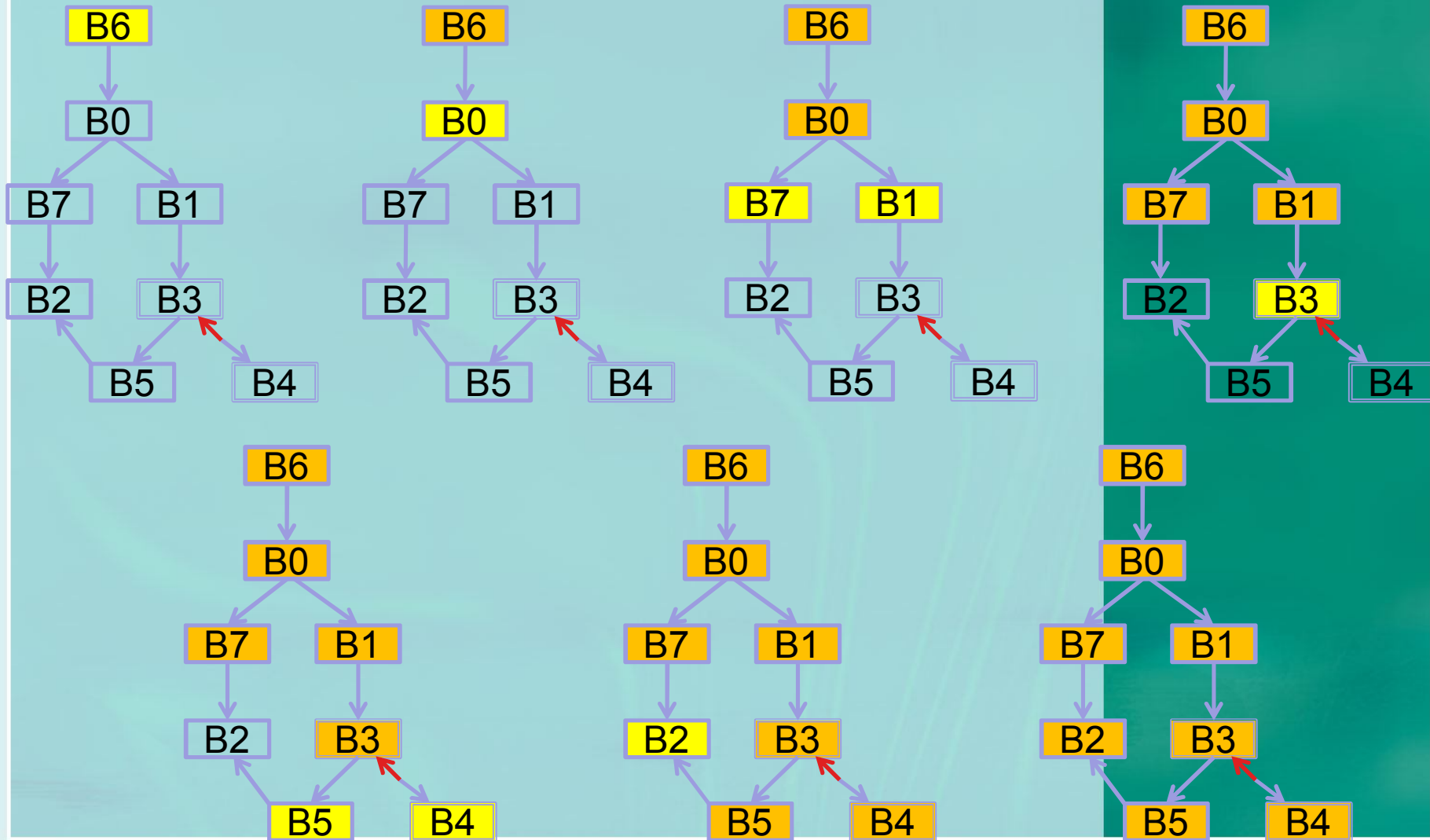
## 2. Loop Positioning Algorithm

- 이 알고리즘은 다른 알고리즘과는 달리 **CFG**를 그리기 위해 특별히 디자인 된 것이다.
- 이 알고리즘의 주요한 목적은 **loop**를 더 세분화 해 준다.
- 가장 바깥쪽 범위는 **loop** 헤더와 같은 **Root Node** 와 같이 취급한다.

# 2-1. Grouping Loops



# 2-2.Modified Breath First Search



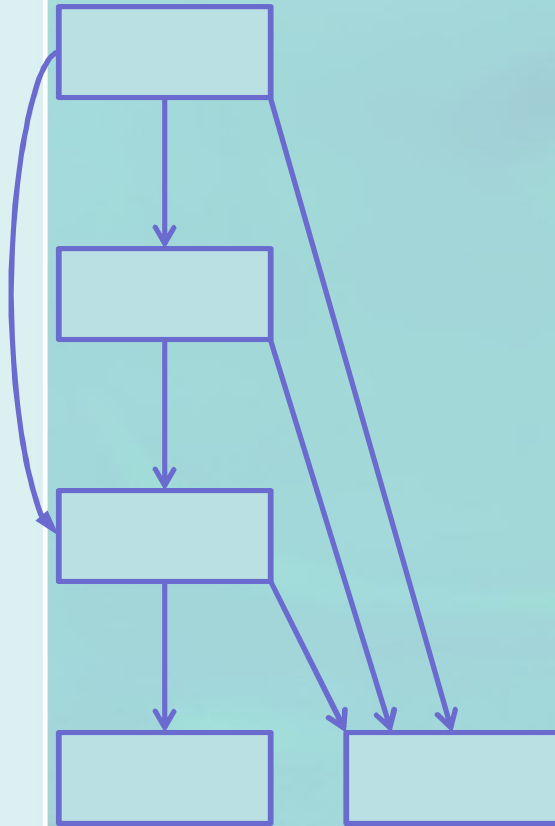


## 3. Hierarchical Positioning Algorithm

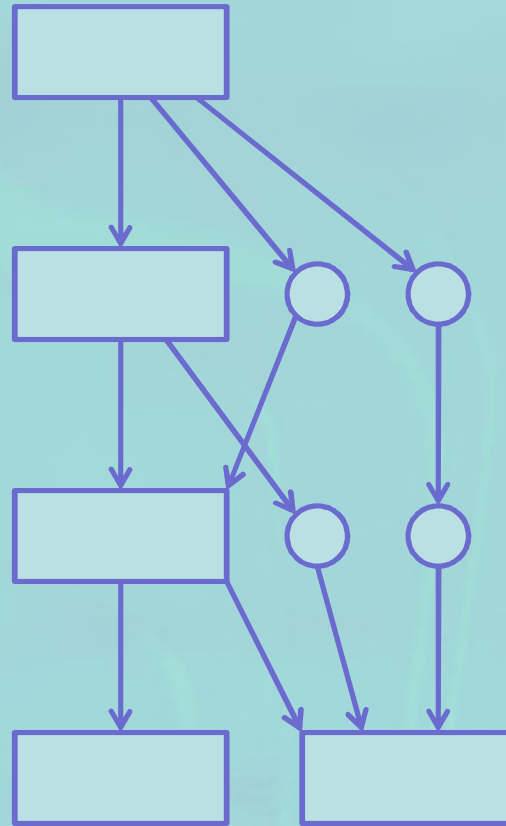
- 이 알고리즘은 directed graph 에서 Node 의 위치를 선정하기 위해 종종 사용 된다.
- 이 알고리즘은 backedge 들을 제거 하고 난 후 acyclic 그래프로 만들고 Dummy Node를 만들어서 서로 엮갈려 있는 edge 들을 풀어 준다.

# 3. Hierarchical Positioning Algorithm Step

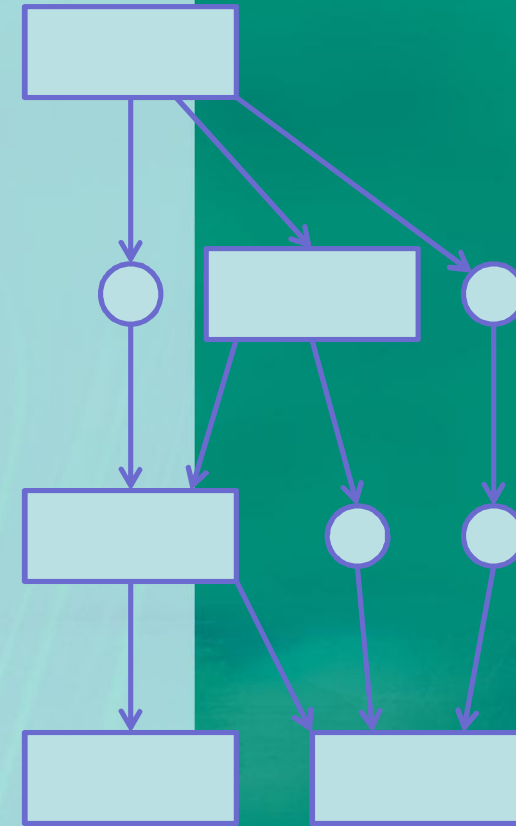
Layer assignment



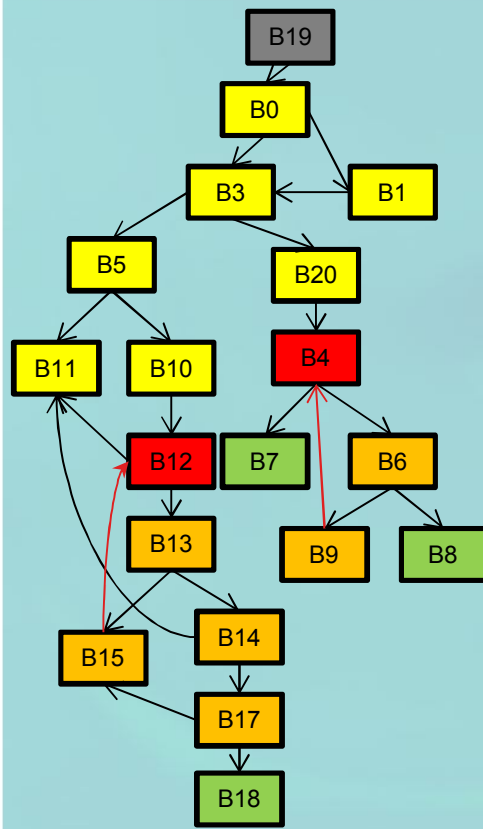
Dummy node insertion



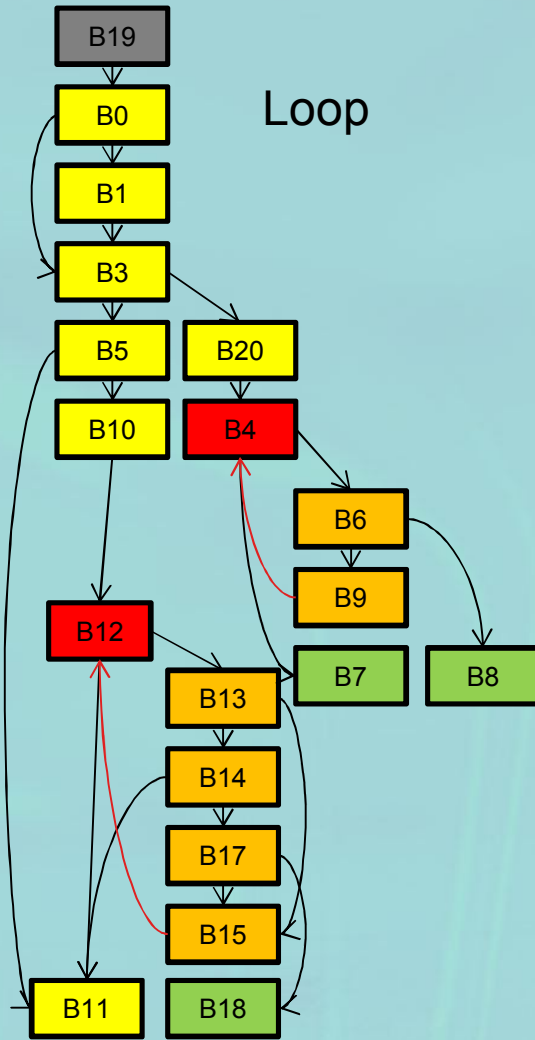
Crossing reduction



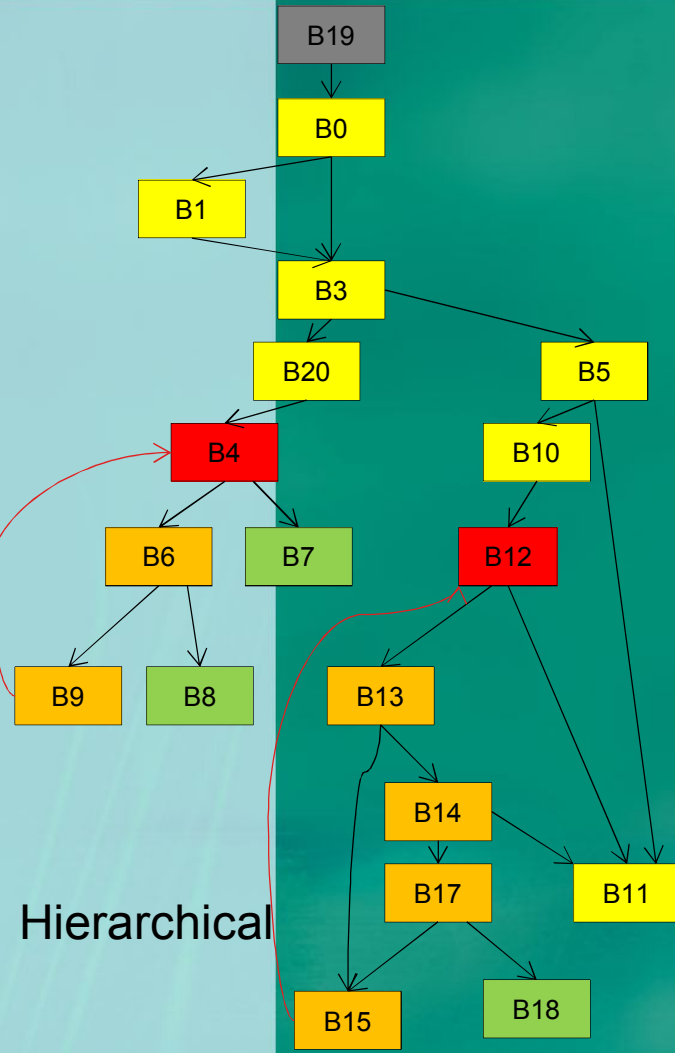
# Comparison



BFS



Loop



Hierarchical

# Statement of Purpose

- 입력 받는 코드는 C코드로 한다.
- 코드를 입력 받아서 CFG를 출력시키는 프로그램이다.
- Loop positioning Algorithm 을 사용하여 CFG를 구현한다
- 시작 Node와 끝 Node 의 색은 회색으로 한다.
- Loop의 header 는 빨강색으로 한다.
- Loop에 속하는 Node중 depth 가 1 이상인 Node 는 주황색으로 표시 한다.
- Child Node 가 없는 Node 는 초록색으로 한다.
- 그 이외의 Node 는 노란색으로 표시한다.
- 일반적인 Edge의 색은 검은색으로 표시한다.
- BackEdge 는 빨간색으로 표시 한다.

## Statement of Purpose(Cont.)

- 각 Node 들은 화살표를 이용하여 route를 나타내고 그 화살표는 다음 Node를 가리킨다.
- 어떠한 Node가 다른 Node들과 아무런 연결도 되어있지 않을 경우에는 제거한다.
- 변환되어 그려진 CFG를 통해서 코드를 최적화할 수 있다.
- 코드가 변환되어 CFG로 그려지는 상세한 방법은 고려하지 않는다.