

4/3 BFS

최백준 choi@startlink.io

미로 탐색

<https://www.acmicpc.net/problem/2178>

- (1, 1) 에서 (N, M)으로 가는 가장 빠른 길을 구하는 문제
- DFS 탐색으로는 문제를 풀 수 없다.
- BFS 탐색을 사용해야 한다.
- BFS는 단계별로 진행된다는 사실을 이용

미로 탐색

<https://www.acmicpc.net/problem/2178>

- (1, 1) 에서 (N, M)으로 가는 가장 빠른 길을 구하는 문제

1	1	0	1	1	0
1	1	0	1	1	0
1	1	1	1	1	1
1	1	1	1	0	1

1					

미로 탐색

<https://www.acmicpc.net/problem/2178>

- (1, 1) 에서 (N, M)으로 가는 가장 빠른 길을 구하는 문제

1	1	0	1	1	0
1	1	0	1	1	0
1	1	1	1	1	1
1	1	1	1	0	1

1	2				
2					

미로 탐색

<https://www.acmicpc.net/problem/2178>

- (1, 1) 에서 (N, M)으로 가는 가장 빠른 길을 구하는 문제

1	1	0	1	1	0
1	1	0	1	1	0
1	1	1	1	1	1
1	1	1	1	0	1

1	2				
2	3				
3					

미로 탐색

<https://www.acmicpc.net/problem/2178>

- (1, 1) 에서 (N, M)으로 가는 가장 빠른 길을 구하는 문제

1	1	0	1	1	0
1	1	0	1	1	0
1	1	1	1	1	1
1	1	1	1	0	1

1	2				
2	3				
3	4				
4					

미로 탐색

<https://www.acmicpc.net/problem/2178>

- (1, 1) 에서 (N, M)으로 가는 가장 빠른 길을 구하는 문제

1	1	0	1	1	0
1	1	0	1	1	0
1	1	1	1	1	1
1	1	1	1	0	1

1	2				
2	3				
3	4	5			
4	5				

미로 탐색

<https://www.acmicpc.net/problem/2178>

- (1, 1) 에서 (N, M)으로 가는 가장 빠른 길을 구하는 문제

1	1	0	1	1	0
1	1	0	1	1	0
1	1	1	1	1	1
1	1	1	1	0	1

1	2				
2	3				
3	4	5	6		
4	5	6			

미로 탐색

<https://www.acmicpc.net/problem/2178>

- (1, 1) 에서 (N, M)으로 가는 가장 빠른 길을 구하는 문제

1	1	0	1	1	0
1	1	0	1	1	0
1	1	1	1	1	1
1	1	1	1	0	1

1	2				
2	3		7		
3	4	5	6	7	
4	5	6	7		

미로 탐색

<https://www.acmicpc.net/problem/2178>

- (1, 1) 에서 (N, M)으로 가는 가장 빠른 길을 구하는 문제

1	1	0	1	1	0
1	1	0	1	1	0
1	1	1	1	1	1
1	1	1	1	0	1

1	2		8		
2	3		7	8	
3	4	5	6	7	8
4	5	6	7		

미로 탐색

<https://www.acmicpc.net/problem/2178>

- (1, 1) 에서 (N, M)으로 가는 가장 빠른 길을 구하는 문제

1	1	0	1	1	0
1	1	0	1	1	0
1	1	1	1	1	1
1	1	1	1	0	1

1	2		8	9	
2	3		7	8	
3	4	5	6	7	8
4	5	6	7		9

미로 탐색

<https://www.acmicpc.net/problem/2178>

- C/C++: <https://gist.github.com/Baekjoon/a9b4e6fa9a984771172d>
- Java: <https://gist.github.com/Baekjoon/3a591ac8f928c56768a62ac8e072fdfb>

숨바꼭질

13

<https://www.acmicpc.net/problem/1697>

- 수빈이의 위치: N
 - 동생의 위치: K
 - 동생을 찾는 가장 빠른 시간을 구하는 문제
-
- 수빈이가 할 수 있는 행동 (위치: X)
 1. 걷기: $X+1$ 또는 $X-1$ 로 이동 (1초)
 2. 순간이동: $2*X$ 로 이동 (1초)

숨바꼭질

<https://www.acmicpc.net/problem/1697>

- 수빈이의 위치: 5
- 동생의 위치: 17
- 5-10-9-18-17 로 4초만에 동생을 찾을 수 있다.

숨바꼭질

15

<https://www.acmicpc.net/problem/1697>

- 큐에 수빈이의 위치를 넣어가면서 이동시킨다
- 한 번 방문한 곳은 다시 방문하지 않는 것이 좋기 때문에, 따로 배열에 체크하면서 방문

숨바꼭질

<https://www.acmicpc.net/problem/1697>

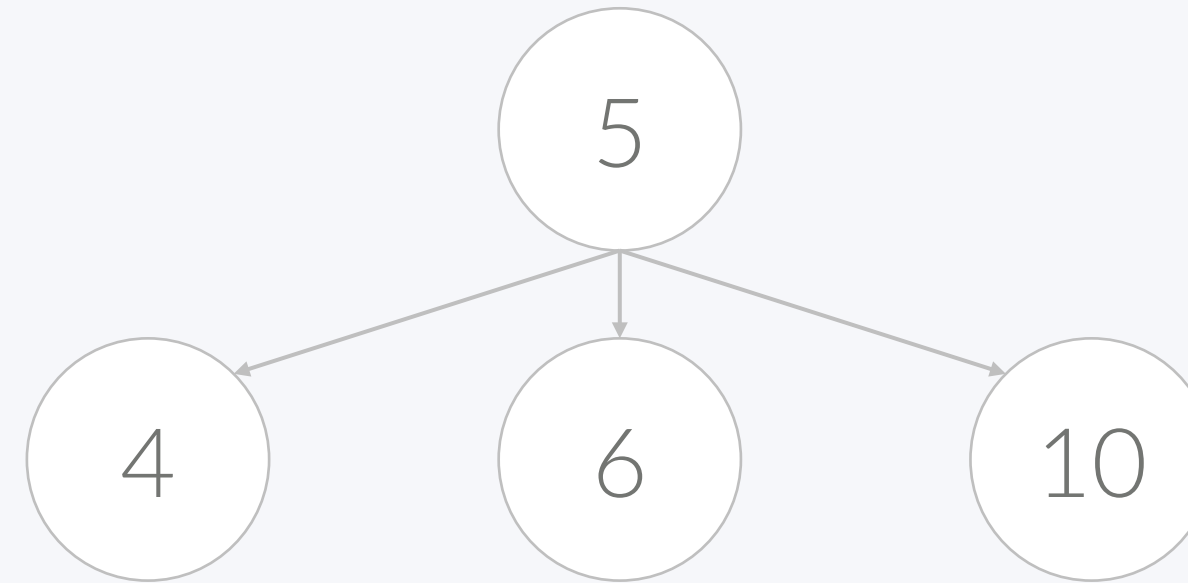
- 가장 처음
- Queue: 5

5

숨바꼭질

<https://www.acmicpc.net/problem/1697>

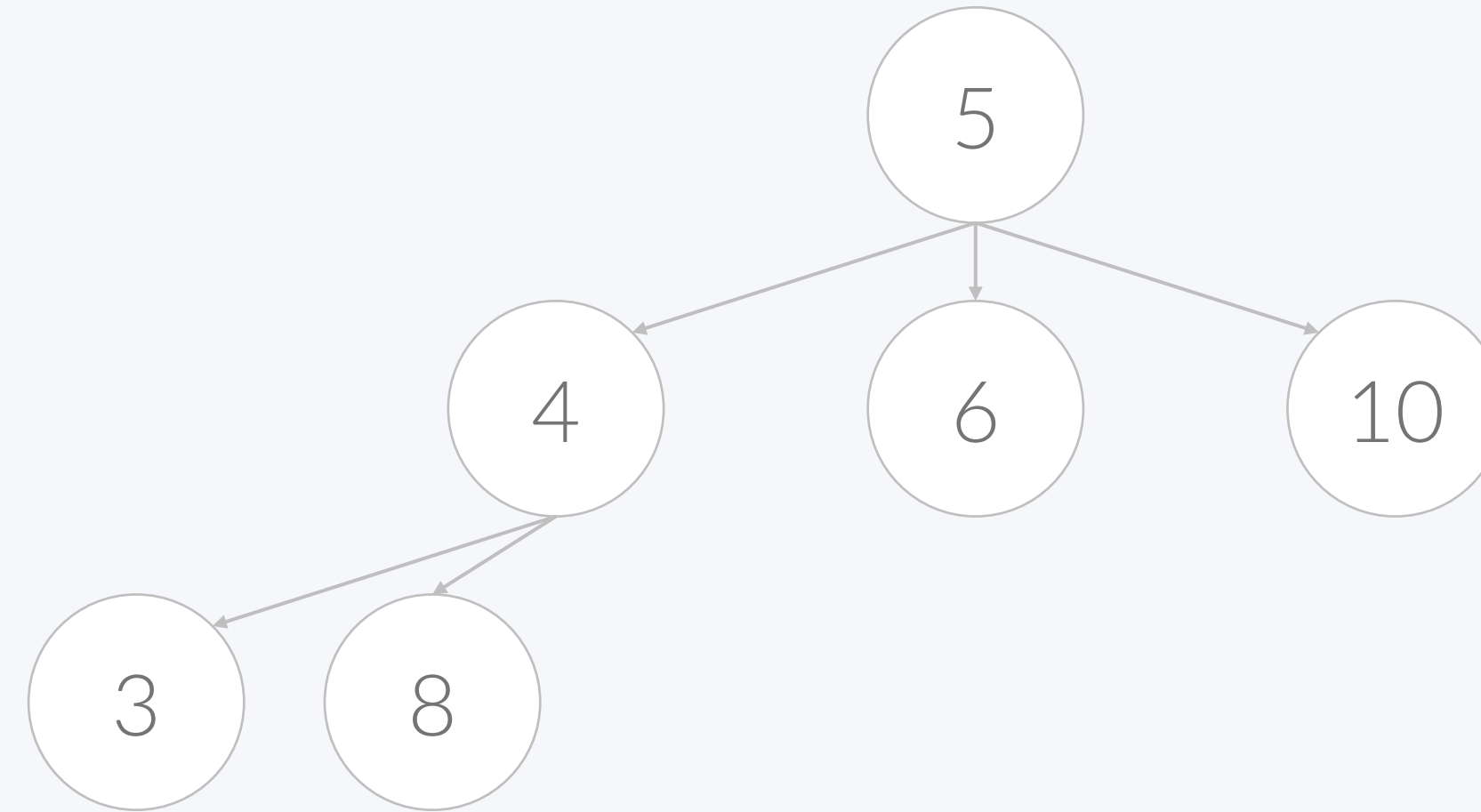
- 5에서 이동
- Queue: 5 4 6 10



숨바꼭질

<https://www.acmicpc.net/problem/1697>

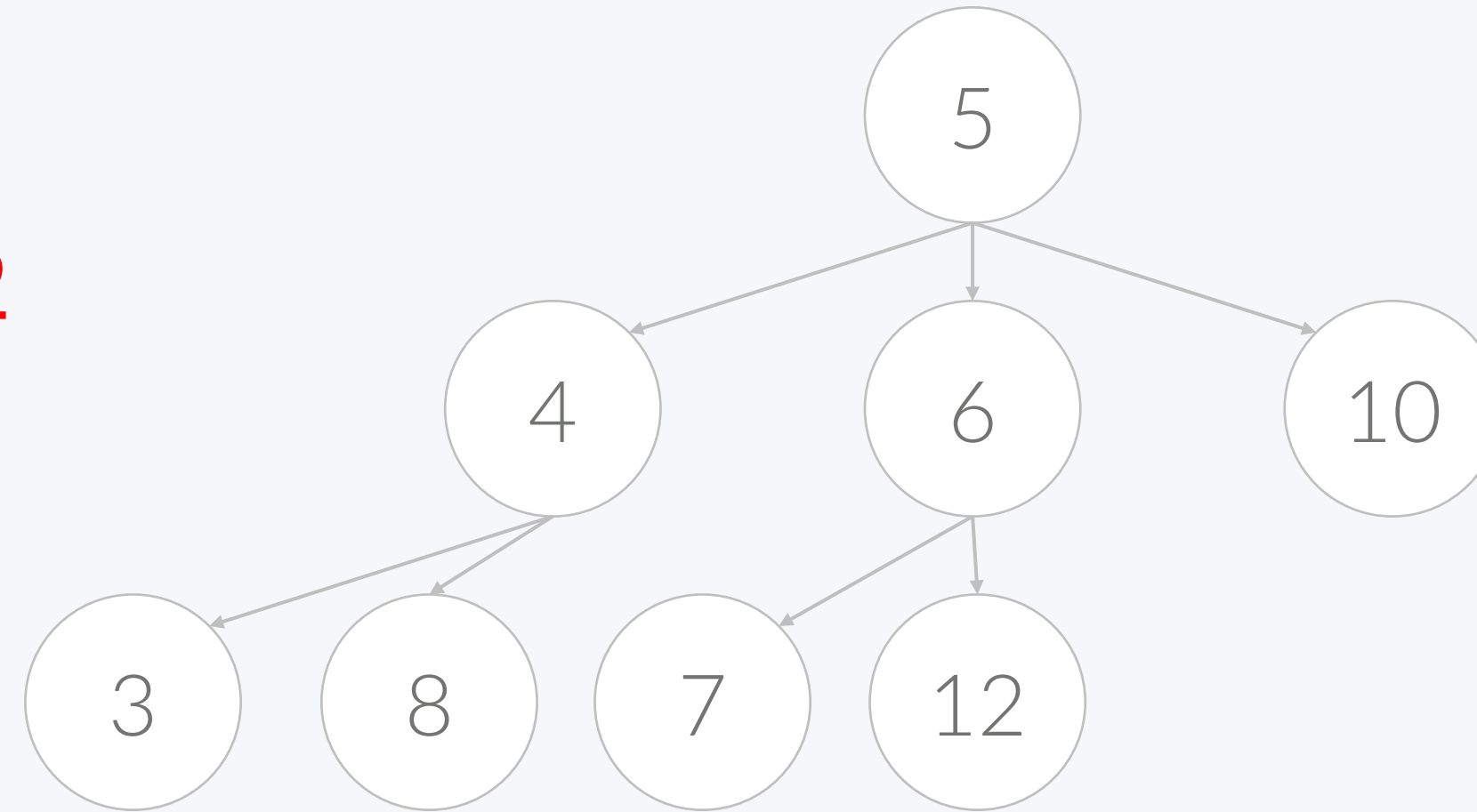
- 4에서 이동
- Queue: 5 4 6 10 3 8



숨바꼭질

<https://www.acmicpc.net/problem/1697>

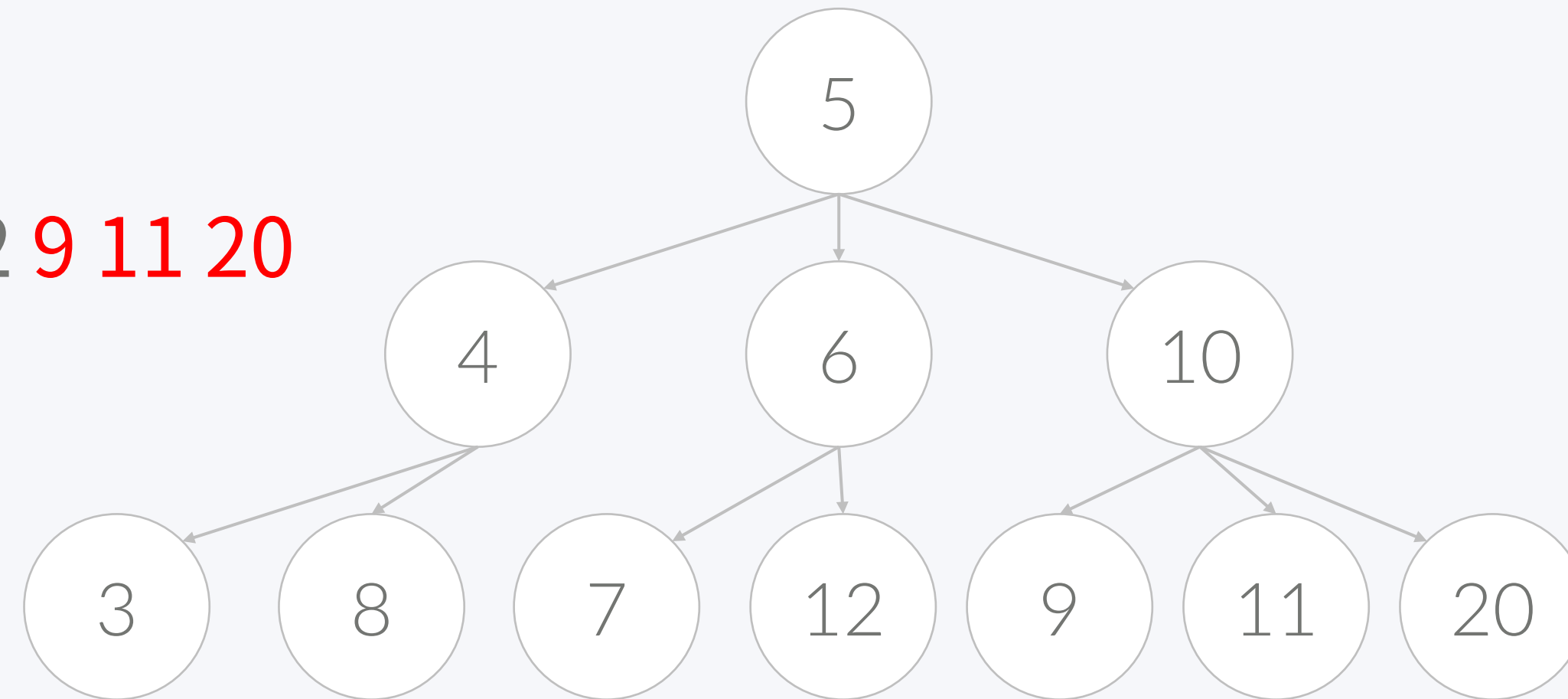
- 6에서 이동
- Queue: 5 4 6 10 3 8 7 12



숨바꼭질

<https://www.acmicpc.net/problem/1697>

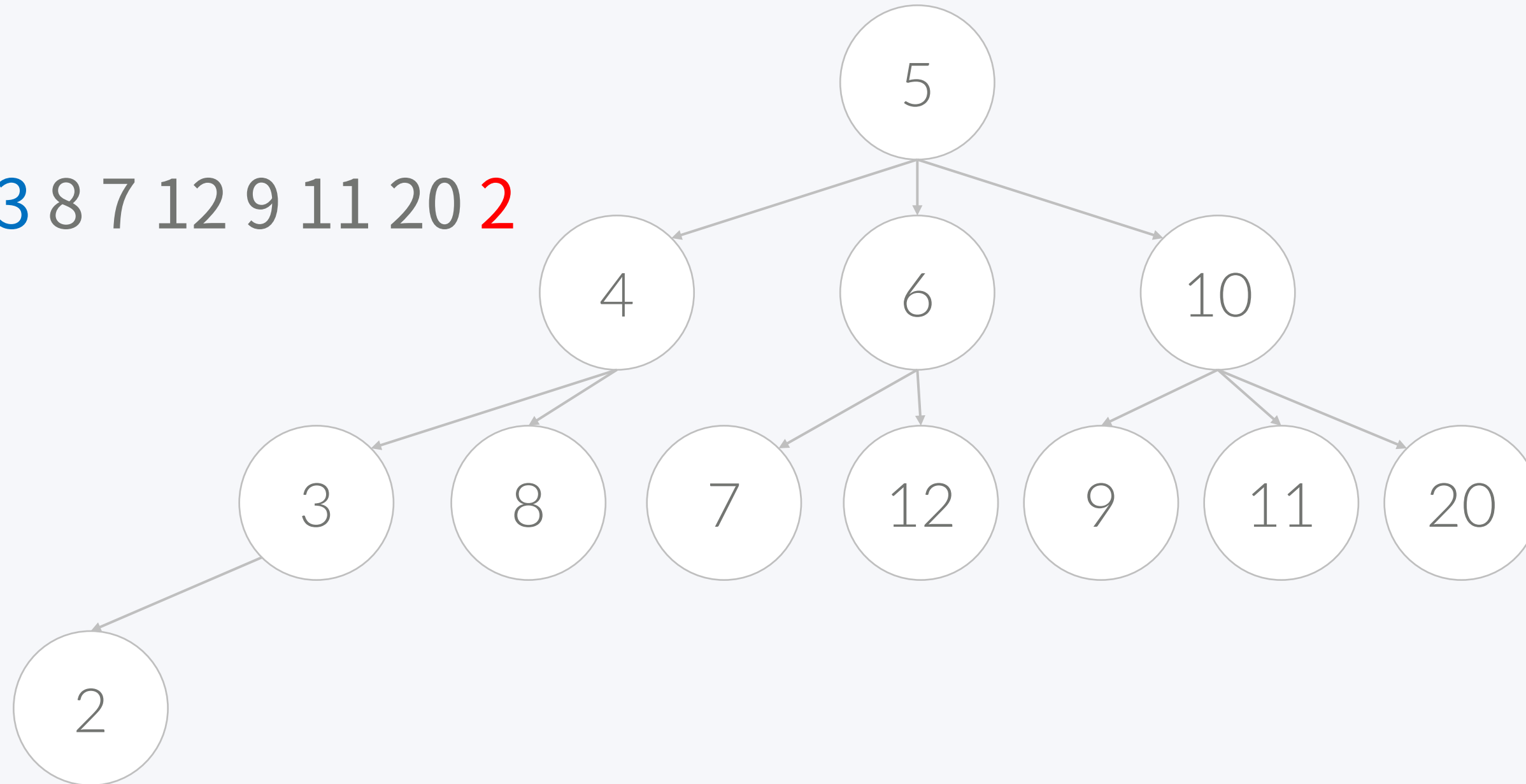
- 10에서 이동
- Queue: 5 4 6 10 3 8 7 12 9 11 20



숨바꼭질

<https://www.acmicpc.net/problem/1697>

- 3에서 이동
- Queue: 5 4 6 10 3 8 7 12 9 11 20 2



숨바꼭질

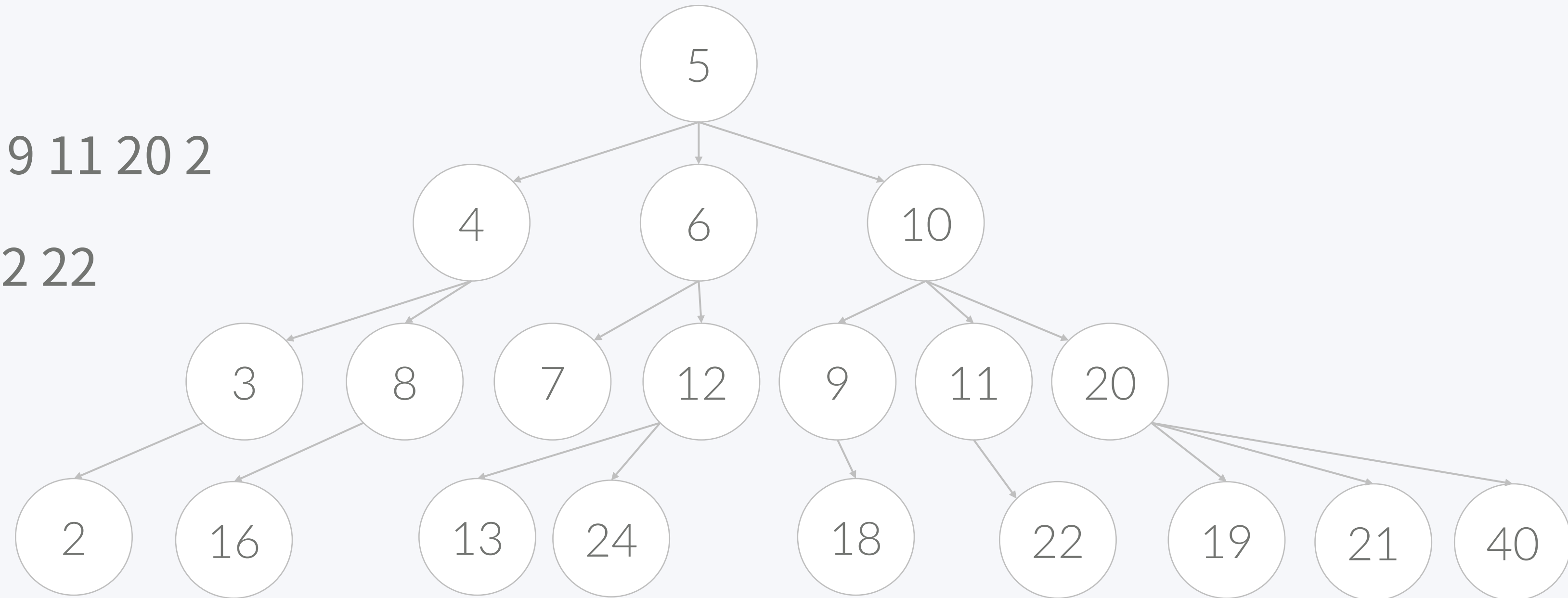
<https://www.acmicpc.net/problem/1697>

- 이런식으로...

숨바꼭질

<https://www.acmicpc.net/problem/1697>

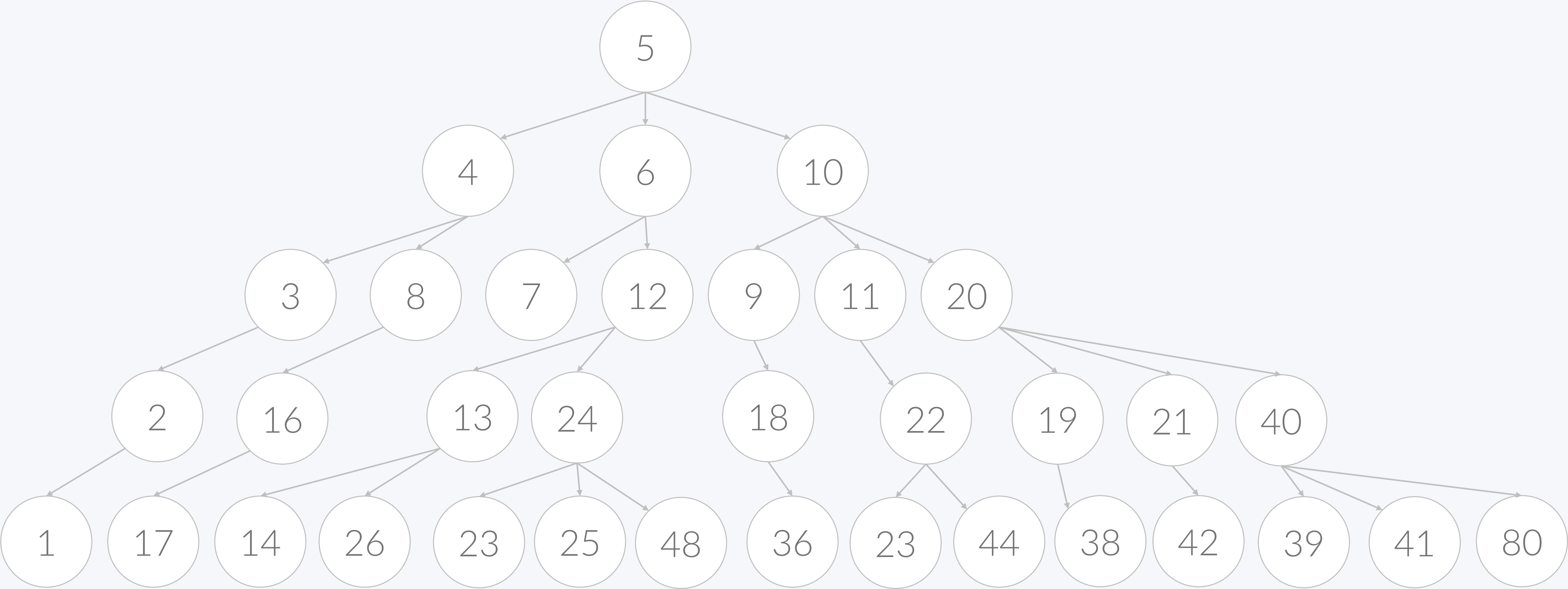
- Queue:
- 5 4 6 10 3 8 7 12 9 11 20 2
- 16 13 15 28 18 12 22
- 19 21 40



숨바꼭질

<https://www.acmicpc.net/problem/1697>

•



숨바꼭질

25

<https://www.acmicpc.net/problem/1697>

- $check[i]$ = i 를 방문했는지
- $dist[i]$ = i 를 몇 번만에 방문했는지

숨바꼭질

26

<https://www.acmicpc.net/problem/1697>

```
check[n] = true;
dist[n] = 0;
queue<int> q;
q.push(n);
while (!q.empty()) {
    int now = q.front();
    q.pop();
    if (now-1 >= 0) {
        if (check[now-1] == false) {
            q.push(now-1);
            check[now-1] = true;
            dist[now-1] = dist[now] + 1;
        }
    }
}
```

```
if (now+1 < MAX) {
    if (check[now+1] == false) {
        q.push(now+1);
        check[now+1] = true;
        dist[now+1] = dist[now] + 1;
    }
}
if (now*2 < MAX) {
    if (check[now*2] == false) {
        q.push(now*2);
        check[now*2] = true;
        dist[now*2] = dist[now] + 1;
    }
}
```

숨바꼭질

<https://www.acmicpc.net/problem/1697>

- C: <https://gist.github.com/Baekjoon/2505b37c55949c9487904f49b5346111>
- C++: <https://gist.github.com/Baekjoon/5b8924d3aec661746358>
- Java: <https://gist.github.com/Baekjoon/e1abcf6deb6796c1282a>

숨바꼭질 4

28

<https://www.acmicpc.net/problem/13913>

- 숨바꼭질 문제 + 이동하는 방법을 출력하는 문제

숨바꼭질 4

<https://www.acmicpc.net/problem/13913>

- now -> next를 갔다고 한다면

```
if (check[next] == false) {  
    q.push(next);  
    check[next] = true;  
    dist[next] = dist[now] + 1;  
}
```

숨바꼭질 4

<https://www.acmicpc.net/problem/13913>

- now -> next를 갔다고 한다면

```
if (check[next] == false) {  
    q.push(next);  
    check[next] = true;  
    from[next] = now;  
    dist[next] = dist[now] + 1;  
}
```

숨바꼭질 4

<https://www.acmicpc.net/problem/13913>

- $\text{from}[i]$ = 어디에서 왔는지
- 의미: $\text{from}[i] \rightarrow i$
- N에서 K를 가는 문제 이기 때문에
- K부터 from을 통해서 N까지 가야한다.
- 즉, 역순으로 저장되기 때문에, 다시 역순으로 구하는 것이 필요하다.

숨바꼭질 4

32

<https://www.acmicpc.net/problem/13913>

```
void print(int n, int m) {  
    if (n != m) {  
        print(n, from[m]);  
    }  
    cout << m << ' ';  
}
```


숨바꼭질 4

<https://www.acmicpc.net/problem/13913>

```
stack<int> ans;
for (int i=m; i!=n; i=from[i]) {
    ans.push(i);
}
ans.push(n);
while (!ans.empty()) {
    cout << ans.top() << ' ';
    ans.pop();
}
cout << '\n';
```

숨바꼭질 4

<https://www.acmicpc.net/problem/13913>

- C: <https://gist.github.com/Baekjoon/3d918ef75f8598f4e73abf0fb5e972ef>
- C++: <https://gist.github.com/Baekjoon/914c1f46da5d4e662758ff6c452dc1e8>
- C++: <https://gist.github.com/Baekjoon/d823eaf585990e75ef5760dd80f232a6>
- Java: <https://gist.github.com/Baekjoon/7dc73e85fe343fc9dbea8b32c1347d7a>
- Java: <https://gist.github.com/Baekjoon/fc8c2fa94562827e8289d59a3fa4b96a>

숨바꼭질 3

35

<https://www.acmicpc.net/problem/13549>

- 수빈이의 위치: N
 - 동생의 위치: K
 - 동생을 찾는 가장 빠른 시간을 구하는 문제
-
- 수빈이가 할 수 있는 행동 (위치: X)
 1. 걷기: $X+1$ 또는 $X-1$ 로 이동 (1초)
 2. 순간이동: $2*X$ 로 이동 (0초)

숨바꼭질 3

36

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

숨바꼭질 3

37

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초 :

5					
---	--	--	--	--	--

• 1초 :

--	--	--	--	--	--

숨바꼭질 3

38

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10				
---	----	--	--	--	--

• 1초:

4	6				
---	---	--	--	--	--

숨바꼭질 3

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10				
---	----	--	--	--	--

• 1초:

4	6				
---	---	--	--	--	--

숨바꼭질 3

40

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10	20			
---	----	----	--	--	--

• 1초:

4	6	9	11		
---	---	---	----	--	--

숨바꼭질 3

41

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

- 0초:

5	10	20			
---	----	----	--	--	--

- 1초:

4	6	9	11		
---	---	---	----	--	--

숨바꼭질 3

42

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

- 0초:

5	10	20			
---	----	----	--	--	--

- 1초:

4	6	9	11	19	
---	---	---	----	----	--

숨바꼭질 3

43

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10	20			
---	----	----	--	--	--

• 1초:

4	6	9	11	19	
---	---	---	----	----	--

• 2초:

--	--	--	--	--	--

숨바꼭질 3

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10	20			
---	----	----	--	--	--

• 1초:

4	6	9	11	19	8
---	---	---	----	----	---

• 2초:

3					
---	--	--	--	--	--

숨바꼭질 3

45

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10	20			
---	----	----	--	--	--

• 1초:

4	6	9	11	19	8						
---	---	---	----	----	---	--	--	--	--	--	--

• 2초:

3					
---	--	--	--	--	--

숨바꼭질 3

46

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10	20			
---	----	----	--	--	--

• 1초:

4	6	9	11	19	8	12					
---	---	---	----	----	---	----	--	--	--	--	--

• 2초:

3	7				
---	---	--	--	--	--

숨바꼭질 3

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10	20			
---	----	----	--	--	--

• 1초:

4	6	9	11	19	8	12					
---	---	---	----	----	---	----	--	--	--	--	--

• 2초:

3	7				
---	---	--	--	--	--

숨바꼭질 3

48

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10	20			
---	----	----	--	--	--

• 1초:

4	6	9	11	19	8	12	18				
---	---	---	----	----	---	----	----	--	--	--	--

• 2초:

3	7	8			
---	---	---	--	--	--

숨바꼭질 3

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10	20			
---	----	----	--	--	--

• 1초:

4	6	9	11	19	8	12	18				
---	---	---	----	----	---	----	----	--	--	--	--

• 2초:

3	7	8			
---	---	---	--	--	--

숨바꼭질 3

50

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10	20			
---	----	----	--	--	--

• 1초:

4	6	9	11	19	8	12	18				
---	---	---	----	----	---	----	----	--	--	--	--

• 2초:

3	7	8			
---	---	---	--	--	--

숨바꼭질 3

51

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10	20			
---	----	----	--	--	--

• 1초:

4	6	9	11	19	8	12	18				
---	---	---	----	----	---	----	----	--	--	--	--

• 2초:

3	7	8			
---	---	---	--	--	--

숨바꼭질 3

52

<https://www.acmicpc.net/problem/13549>

- 5에서 17을 가는 경우 0 ~ 20까지만 위치가 있다고 가정

• 0초:

5	10	20			
---	----	----	--	--	--

• 1초:

4	6	9	11	19	8	12	18	16			
---	---	---	----	----	---	----	----	----	--	--	--

• 2초:

3	7	8			
---	---	---	--	--	--

숨바꼭질 3

<https://www.acmicpc.net/problem/13549>

- 이런식으로 BFS를 진행한다.

숨바꼭질 3

54

<https://www.acmicpc.net/problem/13549>

- 덱을 사용해 순간 이동은 덱의 앞에, 걷기는 덱의 뒤에 넣는 방법도 생각해 볼 수 있다.

숨바꼭질 3

<https://www.acmicpc.net/problem/13549>

- C++ (큐): <https://gist.github.com/Baekjoon/9a2f48a57c0a2782869e4b69026a0972>
- C++ (덱): <https://gist.github.com/Baekjoon/8e524d4a7cfd61263c94e3c6666f99b1>
- Java (큐): <https://gist.github.com/Baekjoon/9134f710bf6d99ff7f1df27e36b208e6>
- Java (덱): <https://gist.github.com/Baekjoon/f83e1a3efaac26be2e254352ee098751>

숨바꼭질 2

<https://www.acmicpc.net/problem/12851>

- 수빈이의 위치: N
 - 동생의 위치: K
 - 동생을 찾는 가장 빠른 시간을 구하는 문제, 그리고 그러한 방법의 개수도 구해야 한다
-
- 수빈이가 할 수 있는 행동 (위치: X)
 1. 걷기: $X+1$ 또는 $X-1$ 로 이동 (1초)
 2. 순간이동: $2*X$ 로 이동 (1초)

숨바꼭질 2

<https://www.acmicpc.net/problem/12851>

```
while (!q.empty()) {
    int now = q.front(); q.pop();
    for (int next : {now-1, now+1, now*2}) {
        if (0 <= next && next <= MAX) {
            if (check[next] == false) {
                q.push(next); check[next] = true;
                dist[next] = dist[now] + 1;
                cnt[next] = cnt[now];
            } else if (dist[next] == dist[now] + 1) {
                cnt[next] += cnt[now];
            }
        }
    }
}
```

숨바꼭질 2

<https://www.acmicpc.net/problem/12851>

- C++: <https://gist.github.com/Baekjoon/39ca797291fc85129c6f1280bf261b5b>
- Java: <https://gist.github.com/Baekjoon/e170e8d3cfef258f001fa951bfa747e7>

이모티콘

<https://www.acmicpc.net/problem/14226>

- 화면에 이모티콘은 1개다
- 할 수 있는 연산
 - 이모티콘을 모두 복사해서 클립보드에 저장
 - 클립보드에 있는 모든 이모티콘을 화면에 붙여넣기
 - 화면에 있는 이모티콘 중 하나를 삭제
- S개의 이모티콘을 치는데 걸리는 시간의 최소값을 구하는 문제

이모티콘

<https://www.acmicpc.net/problem/14226>

- 화면에 이모티콘의 개수 s 와 클립보드에 있는 이모티콘의 개수 c 가 중요하다
- 복사: $(s, c) \rightarrow (s, s)$
- 붙여넣기: $(s, c) \rightarrow (s+c, c)$
- 삭제: $(s, c) \rightarrow (s-1, c)$
- $2 \leq S \leq 1,000$ 이기 때문에 BFS 탐색으로 가능하다.

이모티콘

<https://www.acmicpc.net/problem/14226>

- C++: <https://gist.github.com/Baekjoon/195fcf8798fe3b5b9ae8e46765734265>
- Java: <https://gist.github.com/Baekjoon/5fb20656b011a9842b8e68a44328a398>

알고스팟

<https://www.acmicpc.net/problem/1261>

- 미로는 $N \times M$ 크기이고, 총 1×1 크기의 방으로 이루어져 있다
- 빈 방은 자유롭게 다닐 수 있지만, 벽은 부수지 않으면 이동할 수 없다
- (x, y) 에 있을 때, 이동할 수 있는 방은 $(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$ 이다
- $(1, 1)$ 에서 (N, M) 으로 이동하려면 벽을 최소 몇 개 부수어야 하는지 구하는 문제

- 처음 상태

0	0	1	1	1	1
0	1	0	0	0	0
0	0	1	1	1	1
1	1	0	0	0	1
0	1	1	0	1	0
1	0	0	0	1	0

[illegible]

알고스팟

<https://www.acmicpc.net/problem/1261>

- 벽을 부수지 않고 이동할 수 있는 곳

0	0	1	1	1	1
0	1	0	0	0	0
0	0	1	1	1	1
1	1	0	0	0	1
0	1	1	0	1	0
1	0	0	0	1	0

0	0				
0					
0	0				

알고스팟

<https://www.acmicpc.net/problem/1261>

- 벽을 1개 부수고 이동할 수 있는 곳

0	0	1	1	1	1
0	1	0	0	0	0
0	0	1	1	1	1
1	1	0	0	0	1
0	1	1	0	1	0
1	0	0	0	1	0

0	0	1			
0	1	1	1	1	1
0	0	1			
1	1	1	1	1	
1		1	1		
	1	1	1		

알고스팟

<https://www.acmicpc.net/problem/1261>

- 벽을 2개 부수고 이동할 수 있는 곳

0	0	1	1	1	1
0	1	0	0	0	0
0	0	1	1	1	1
1	1	0	0	0	1
0	1	1	0	1	0
1	0	0	0	1	0

0	0	1	2	2	2
0	1	1	1	1	1
0	0	1	2	2	2
1	1	1	1	1	2
1	2	1	1	2	2
2	1	1	1	2	2

알고스팟

<https://www.acmicpc.net/problem/1261>

- BFS탐색을 벽을 부순 횟수에 따라서 나누어서 수행해야 한다.
- C/C++: <https://gist.github.com/Baekjoon/3d8ed2a3976c7affbd73>
- Java: <https://gist.github.com/Baekjoon/e66c8bed6b5d440d5bafbf882acc19e>
- 시간 복잡도: $O(N^2)$

알고스팟

<https://www.acmicpc.net/problem/1261>

- 어차피 벽을 뚫는다고 안 뚫는다고 나누어지기 때문에, 덱을 사용한다
- 벽을 뚫는 경우에는 뒤에, 안 뚫는 경우에는 앞에 추가한다.
- C/C++: <https://gist.github.com/Baekjoon/9da1eed82383645026cc>
- Java: <https://gist.github.com/Baekjoon/0cc05c42a1ab3b3b325f50e1de371442>
- 시간 복잡도: $O(N^2)$

벽 부수고 이동하기

<https://www.acmicpc.net/problem/2206>

- $N \times M$ 의 행렬로 나타내는 지도에서 (1, 1)에서 (N,M)으로 최단 거리로 이동하는 문제
- 0은 빈 칸, 1은 벽
- 단, 벽은 한 번 부수고 지나갈 수 있다

벽 부수고 이동하기

<https://www.acmicpc.net/problem/2206>

- 벽을 부순다는 조건이 없으면 일반적인 미로 탐색 문제이다
- 어떤 칸에 방문했을 때, 벽을 부순 적이 있는 경우와 아직 부순 적이 없는 경우는 다른 경우이기 때문에
- 상태 (i, j) 대신에 (i, j, k) ($k == 0$ 이면 벽을 부순 적이 없음, 1이면 있음) 으로 BFS 탐색을 진행한다.

벽 부수고 이동하기

<https://www.acmicpc.net/problem/2206>

- C++: <https://gist.github.com/Baekjoon/488504115d1acaf9dc319d05c4b59418>
- Java: <https://gist.github.com/Baekjoon/ac92846bffe63f6f8c314c5415b9b7>

탈출

<https://www.acmicpc.net/problem/3055>

- 지도는 R행 C열이다
- 비어있는 곳은 '.'
- 물이 차있는 지역은 '*'
- 돌은 'X'
- 비버의 굴은 'D'
- 고슴도치의 위치는 'S'

탈출

<https://www.acmicpc.net/problem/3055>

- 먼저, 물이 언제 차는지 미리 구해놓은 다음에
- 고슴도치를 그 다음에 이동시킨다

탈출

74

<https://www.acmicpc.net/problem/3055>

- 지도 상태

.	D	.	*
.	.	.	.
.	.	X	.
S	.	*	.
.	.	.	.

- 물이 차는 시간

5	-1	1	0
4	3	2	1
3	2	-1	0
2	1	0	1
3	2	1	2

- 고슴도치의 이동

0			

탈출

<https://www.acmicpc.net/problem/3055>

- 지도 상태
- 물이 차는 시간
- 고슴도치의 이동

.	D	.	*
.	.	.	.
.	.	X	.
S	.	*	.
.	.	.	.

5	-1	1	0
4	3	2	1
3	2	-1	0
2	1	0	1
3	2	1	2

1			
0			
1			

탈출

<https://www.acmicpc.net/problem/3055>

- 지도 상태
- 물이 차는 시간
- 고슴도치의 이동

.	D	.	*
.	.	.	.
.	.	X	.
S	.	*	.
.	.	.	.

5	-1	1	0
4	3	2	1
3	2	-1	0
2	1	0	1
3	2	1	2

2			
1			
0			
1			

탈출

<https://www.acmicpc.net/problem/3055>

- 지도 상태
- 물이 차는 시간
- 고슴도치의 이동

.	D	.	*
.	.	.	.
.	.	X	.
S	.	*	.
.	.	.	.

5	-1	1	0
4	3	2	1
3	2	-1	0
2	1	0	1
3	2	1	2

3			
2			
1			
0			
1			

탈출

<https://www.acmicpc.net/problem/3055>

- 지도 상태
- 물이 차는 시간
- 고슴도치의 이동

.	D	.	*
.	.	.	.
.	.	X	.
S	.	*	.
.	.	.	.

5	-1	1	0
4	3	2	1
3	2	-1	0
2	1	0	1
3	2	1	2

3	4		
2			
1			
0			
1			

탈출

<https://www.acmicpc.net/problem/3055>

- C++: <https://gist.github.com/Baekjoon/f540851ce65619a8b07c65ac3f2ca245>
- Java: <https://gist.github.com/Baekjoon/cd2de8bd2b931622f304df1a2a6c2f1b>

탈옥

<https://www.acmicpc.net/problem/9376>

- 빈 칸, 벽, 문으로 이루어진 지도가 주어진다.
- 두 죄수가 탈옥하기 위해서 열어야 하는 문의 최소 개수를 구하는 문제

탈옥

<https://www.acmicpc.net/problem/9376>

- 두 지도를 상하좌우로 한 칸씩 확장하면
- 두 죄수의 탈옥 경로는
- 어딘가에서 만나서 함께 이동하는 꼴이 된다
- 따라서, 지도의 밖에서 BFS 1번, 각 죄수별로 1번씩 BFS를 수행한다.
- 그 다음, 정답을 합친다
- 이 때, 문에서 만나는 경우는 조심해야 한다

[illegible]

<https://www.acmicpc.net/problem/9376>

밖에서 부터

[illegible]

죄수 1부터

[illegible]

죄수 2부터

[illegible]

죄수 2부터

[illegible][illegible][illegible]

<https://www.acmicpc.net/problem/9376>

- C++: <https://gist.github.com/Baekjoon/682466914543f0aeccc3da8941c015d1>
- Java: <https://gist.github.com/Baekjoon/99c1122a37a2e025b1994eb52342a35f>

열쇠

<https://www.acmicpc.net/problem/9328>

- BFS를 큐 27개로 수행해야 한다.
- 큐 1개: 일반적인 BFS
- 큐 26개: 문을 열기 위해 기다리는 큐

열쇠

<https://www.acmicpc.net/problem/9328>

- C++: <https://gist.github.com/Baekjoon/a2442e638027d8c8174c>
- Java: <https://gist.github.com/Baekjoon/c5f9a4756842bcbd8da639ae89f5af7e>

발레리노

<https://www.acmicpc.net/problem/1857>

- m행 n열 격자판이 주어진다.
- 김주성은 체스판의 나이트와 같이 이동한다.
 - 0: 맨땅
 - 1: 방석이 깔려 있음
 - 2: 돌맹이가 있음 (이동할 수 없음)
 - 3: 김주성의 시작 위치 (방석 있음)
 - 4: 김주성의 도착 위치 (방석 있음)
- 3 -> 4로 가기 위해 필요한 방석의 최소 개수와 놓는 방법의 수

발레리노

<https://www.acmicpc.net/problem/1857>

- 일부 정점이 색칠되어져 있는 그래프가 주어진다.
- s 에서 t 로 가기 위해 최소 몇 개의 정점을 색칠해야 하는가?

발레리노

<https://www.acmicpc.net/problem/1857>

- 그래프에서 최소 비용과 최소 비용이 나오는 방법의 수를 구하는 문제이다

발레리노

<https://www.acmicpc.net/problem/1857>

- 방석을 놓지 않고 갈 수 있는 부분을 정점 하나로 그룹을 지었다고 생각하면
- 이 문제는, s 에서 t 로가는 가장 빠른 길을 구하는 문제가 되어버린다.

발레리노

<https://www.acmicpc.net/problem/1857>

- C++: <https://gist.github.com/Baekjoon/39dbbef318bb62f4da19b1e57943b4fc>
- Java: <https://gist.github.com/Baekjoon/28e92ba0cf053e9f68b6b39294ea8abf>