Chapter 1
알고리즘의 첫걸음

차례

알고리즘의 유래

- 알고리즘

  9세기경 페르시아 수학자인 알콰리즈미(al Khwarizmi)의 이름으로부터 유래

- 최초의 알고리즘

  BC 300년 경 유클리드의 최대공약수(GCD) 알고리즘

그림 출처: http://en.wikipedia.org/wiki/Mu%E1%B8%A5ammad_ibn_M%C5%ABs%C4%81_al-Khw%C4%81rizm%C4%AB

알고리즘이란?

- 알고리즘

  문제를 해결하기 위한 단계적인 절차를 의미

- 알고리즘은 요리법과 유사

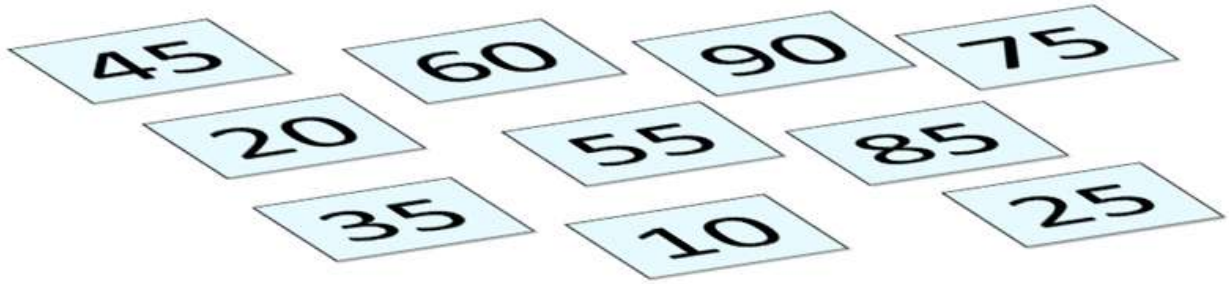
  단계적인 절차를 따라 하면 요리가 만들어지듯이, 알고리즘도 단계적인 절차를 따라 하면 주어진 문제의 해를 찾는다.

- 효율적인 알고리즘 고안이 중요

  주어진 문제에 대해 여러 종류의 알고리즘이 있을 수 있으나, 항상 보다 효율적인 알고리즘을 고안하는 것이 매우 중요

1.1 최대 숫자 찾기

➢ 가장 큰 숫자 찾기



알고리즘

카드의 숫자를 하나씩 비교하면서 본 숫자들 중에서 가장 큰 숫자를 기억해가며 찾아보자.

- 마지막 카드의 숫자를 본 후에, 머릿속에 기억된 가장 큰 숫자가 적힌 카드를 바닥에서 집어 든다.

- 이런 방법을 **순차 탐색** (Sequential Search)이라 한다.

  즉, 카드를 한 장씩 차례대로 (주어진 순서대로) 읽어 가며 찾는 방법

1.2 임의의 숫자 찾기

- 85를 찾아보자.

45	20	60	35	10	55	90	85	75	25

알고리즘

최대 숫자 찾기처럼 85를 기억하고 바닥에 펼쳐진 카드를 차례대로 한 장씩 읽으며 85와 비교

⇨ 순차 탐색

문제

- 10장의 카드가 오름차순으로 미리 정렬되어 있다면?

실생활의 예:

사전, 휴대폰의 전화번호 리스트, 책 뒷부분의 인덱스 등

- 85 찾기

15, 20, 25, 35, 45, 55, 60, 75, 85, 90에서 85를 순차 탐색으로 찾으면

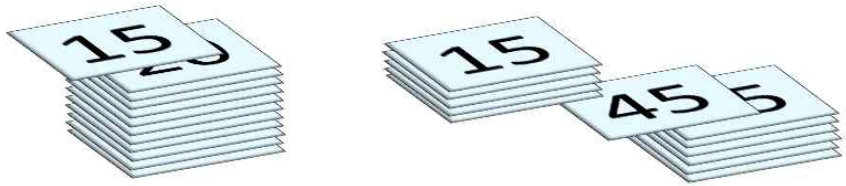
위로부터 8장의 카드를 읽은 후에나 85를 찾는다.

순차 탐색보다 효율적인 방법은?

- 정렬되어 있다는 정보를 어떻게 활용할 수 있을까?

중간에 있는 45 (혹은 55)와 85를 먼저 비교

15, 20, 25, 35, 45, 55, 60, 75, 85, 90



중간 한 장을 읽어 85와 비교한 것이 첫 카드부터 읽어 나가는 순차 탐색보다 훨씬 빠르게 목표(85)에 다가간다.

➢ 이진 탐색 (Binary Search), K 찾기

오름차순으로 데이터를 정렬한다.

중간 숫자와 K 비교한 후

같으면 탐색 성공

K가 작으면 앞부분 반에서 같은 방법으로 K를 찾고

K가 크면 뒷부분 반에서 같은 방법으로 K를 찾는다.

문제

1.3 동전 거스름돈

- 물건을 사고 거스름돈을 동전으로 받아야 한다면?

  대부분의 경우 가장 적은 수의 동전을 원한다.

- 거스름돈이 730원이라면?

  500원 동전 1개, 100원 동전 2개, 10원 동전 3개인 총 6개를 거슬러 받으면

  거스름돈 730원에 대한 최소 동전의 수

어떻게 하면 가장 적은 수의 동전을 찾을까?

가장 큰 액면의 동전부터 차례로

- 730원 대해서

  500원 동전 1개를 선택

  남은 230원에 대해 100원 동전 2개 선택

  남은 30원에 대해 10원 동전 3개를 선택

  총 동전의 수 = 1 + 2 + 3 = 6개 (최소 동전의 수)

남은 거스름돈 액수를 넘지 않는 한도에서 (욕심 내어) 가장 큰 액면의 동전을 계속하여 선택

이러한 종류의 알고리즘을 그리디(Greedy) 알고리즘이라고 한다.

1.4 한붓그리기

➢ 종이에서 연필을 떼지 않고 그리는 한붓그리기 문제

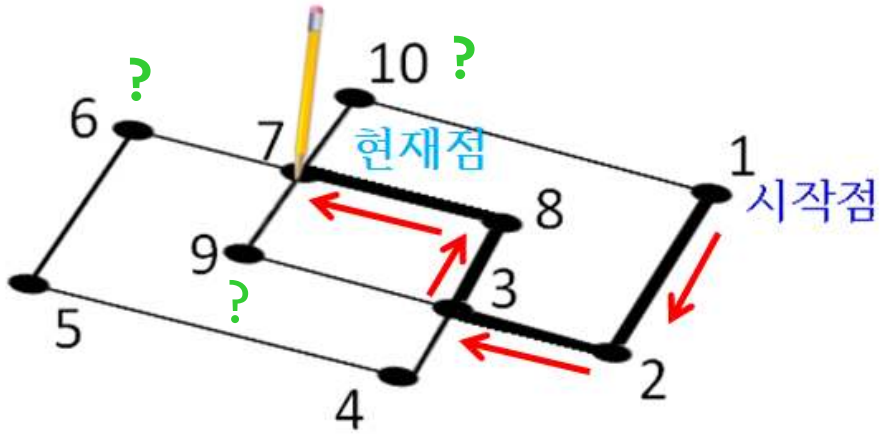
어느 한 점에서 출발하여 모든 간선을 한 번만 지나서 출발점으로 돌아오되, 궤적을 그리는 동안 연필이 종이에서 떨어져서는 안된다.

단, 점은 여러 차례 방문하여도 무방하다.

어떻게 해결 방법을 찾아야 할까?

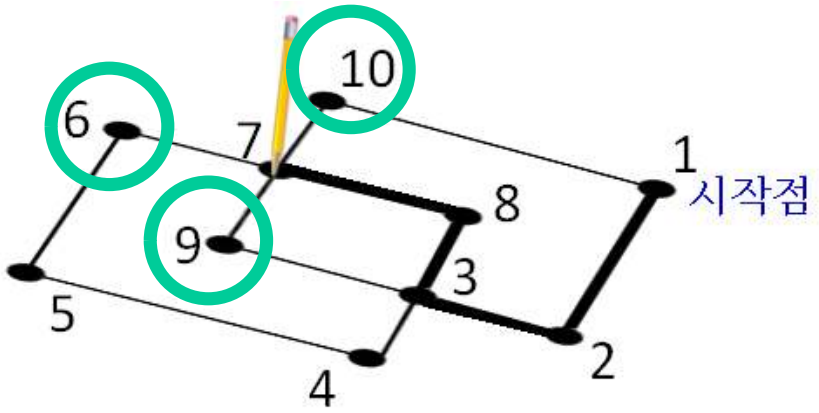
그래프가 작으면 연필로 시행착오를 통해서

그래프가 크면 그래프의 어느 점까지 진행하여 왔을 때, 다음에 어느 점으로 이동해야 할지를 결정하기가 쉽지 않다.



현재 점 (점 7)으로부터 점 6, 9, 또는 10 중에서 어디로 진행해야 할까?

점 6, 9, 10을 살펴보자!



- 점 6으로 가면, 5, 4, 3, 9, 7, 10을 거쳐서 점 1로 돌아올 수 있다.
- 점 9로 가면, 3, 4, 5, 6, 7, 10을 거쳐서 점 1로 역시 돌아올 수 있다.
- 점 10으로 가면, 점 1로 갈 수 밖에 없고, 3, 4, 5, 6, 7, 9 사이의 간선을 지나가기 위해서는 연필을 떼어야 한다.

➢ 점 6, 9의 공통점은?

점 6, 9의 공통점은?

- 현재 점으로부터 점 6이나 점 9를 지나서 현재 점으로 돌아오는 사이클이 있다.
  - 즉, 현재 점에서 점 6이나 점 9로 이동하면, 연필을 떼지 않고 진행할 수 있다.

- 따라서, 점 6, 9의 공통점은?

  현재 점에서 현재 점으로 돌아오는 사이클이다.

알고리즘

- 현재 점으로 돌아오는 사이클이 있으면 진행한다.
- 단, 외길이면, 즉, 인접한 점이 하나밖에 없으면 사이클 체크 없이 인접한 점으로 진행한다.

1.5 미로 찾기

➢ 그리스 신화

지중해의 크레타 섬을 통치하던 폭군 미노스 (Minos) 왕은 황소 머리에 하반신은 사람인 무서운 짐승 미노타우르에게 제물을 바치기 위해 아테네에 젊은 남녀를 조공으로 요구

조공으로 바쳐진 젊은이들은
지하에 있는 매우 광대하고
복잡한 미로 내부에서 탈출하지
못하여 미노타우르에게 잡아먹혔다.

출처: 위키피디아

아테네의 한 젊은 청년 테세우스(Theseus)가 나섰다.

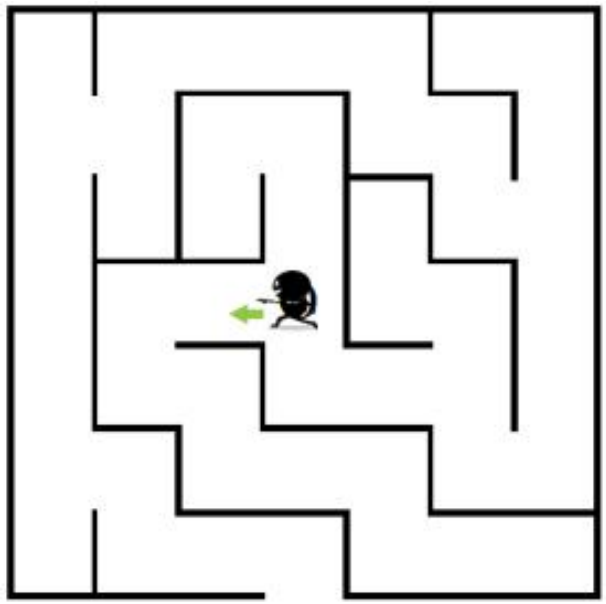
그는 다행히 칼과 함께 실타래를 가지고 실을 풀면서 미로에 들어갔다.

그리고 마침내 테세우스는 미노타우르를 칼로 죽이고, 실을 다시 감으면서 미로를 빠져 나왔다.

그리스 신화에서 알려주는 미로 찾기의 해는 바로 실타래이다.

실제 미로 문제

- 대부분의 경우는 실타래가 없다.
- 이러한 상황에서 어떻게 미로를 빠져나올 수 있을까?



➢ 일반적인 방법

현 위치에서 한 방향을 선택하여 이동하고,

길이 막혀 있으면 다시 되돌아 나오며,

다른 방향으로 다시 시도해 본다.

가능하다면 지나갔던 곳을 표시해 놓고 다시 안 간다.

- 그러나 이 방법은 그리스 신화 속의 미로에서는 불가능하다.

  지하의 미로에서는 어두워서 표시한 곳을 식별할 수 없다.

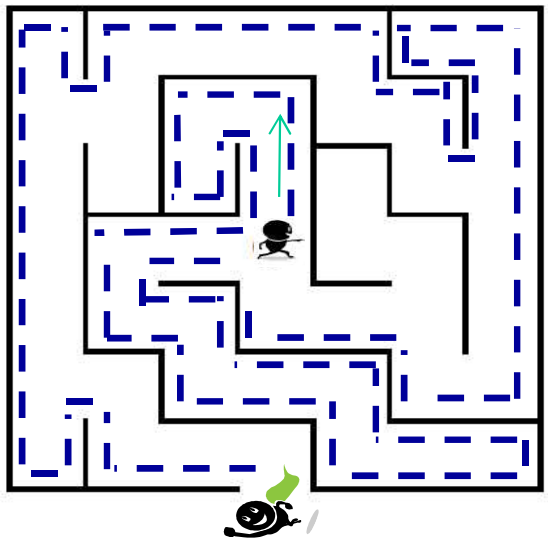
- 이렇게 어려운 상황에서도 답은 존재한다. 무엇일까?

➢ 오른손 법칙

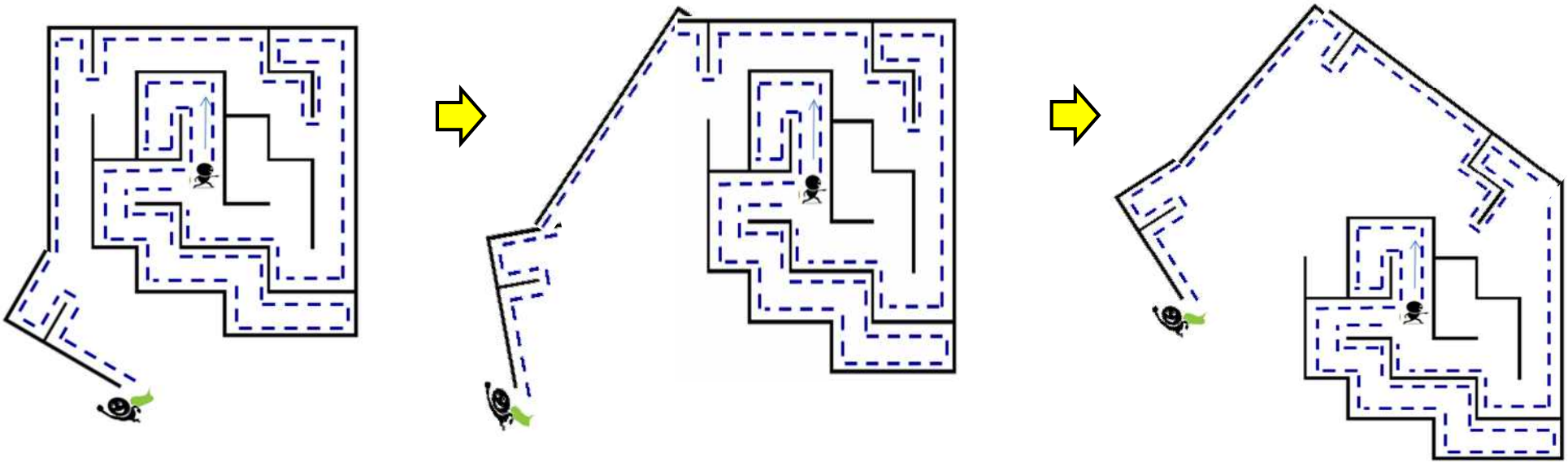
현 위치에서 한 방향을 선택하고, 오른손을 벽에 댄다.

그리고 출구가 나올 때까지 계속 오른손을 벽에서 떼지 않고 계속 걸어간다.

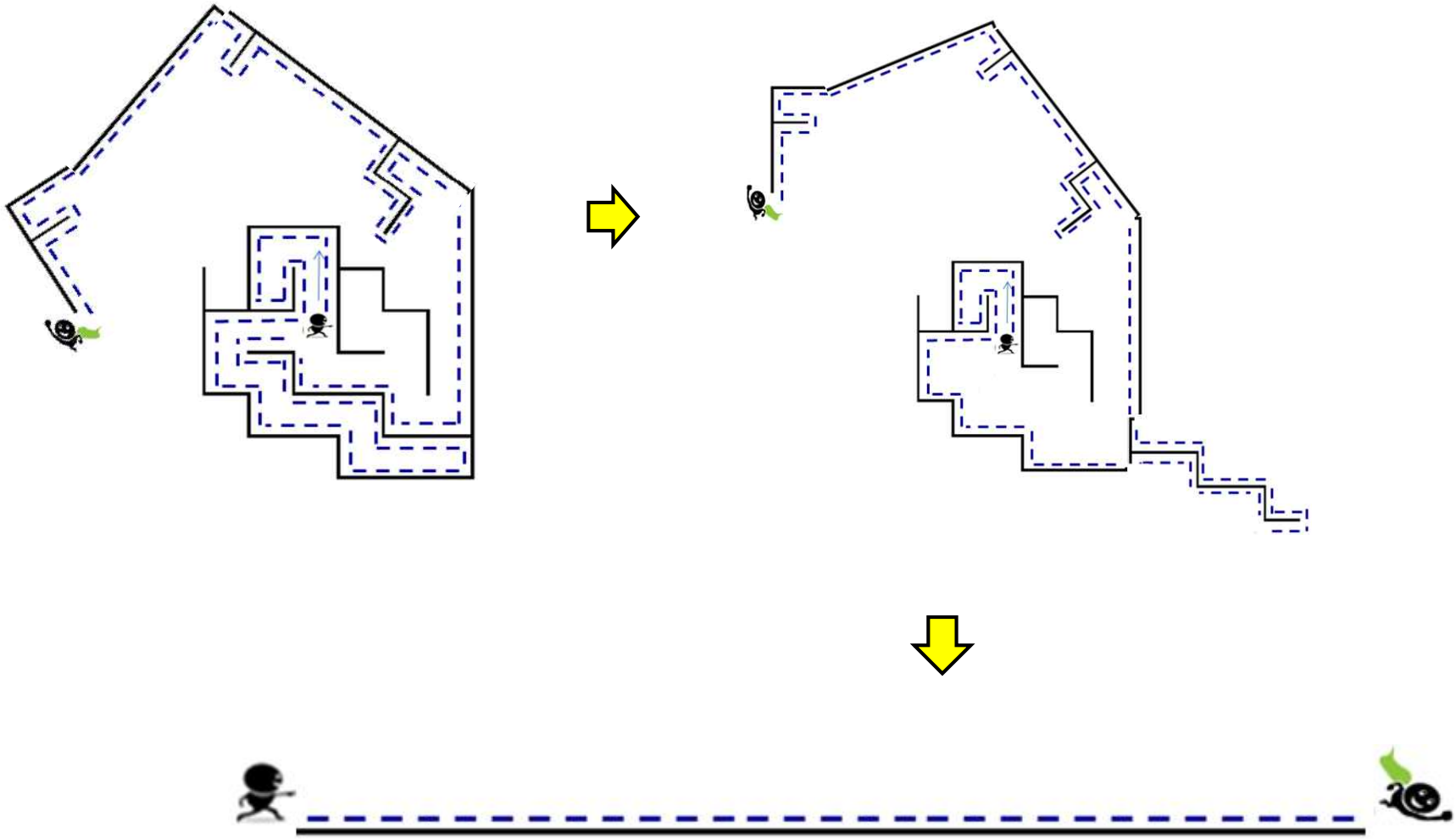
이 방법은 실타래나 특별한 표시가 필요 없이도 항상 출구를 찾는다.



오른손 법칙으로 항상 출구를 찾는 이유



미로가 종이 박스로 만들어져 있다고 생각해보면



1.6 가짜 동전 찾기

- 아주 많은 동전 속에 1개의 가짜 동전이 있다.

  가짜 동전은 눈으로 식별할 수 없다.

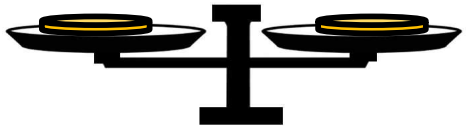
  그러나 가짜 동전의 무게는 정상적인 동전보다 약간 가볍다.

- 가짜 동전 찾기 문제

  최소의 양팔 저울을 사용하는 횟수로 가짜 동전을 찾아내기

철수 아이디어

동전 1개를 저울 왼편에 올리고, 나머지 동전을 하나씩 오른편에 올려서 가짜를 찾자.



- 1 ~ (n-1)회

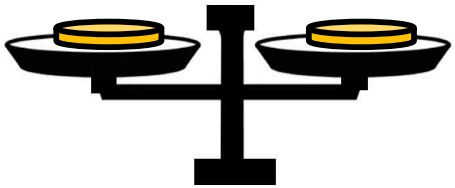
  운이 좋으면 1번 만에 가짜 동전을 찾는다.

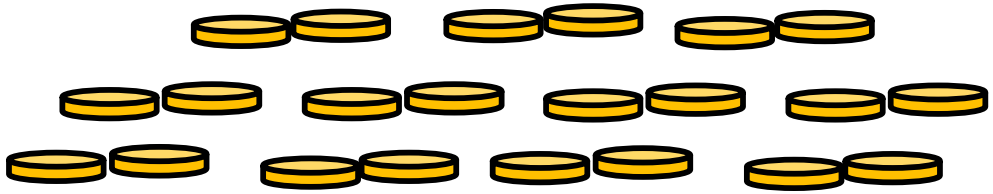
  최악은 가장 마지막에 가짜 동전을 올려놓게 되는 경우

  총 동전 수가 n이라면 (n-1)번 저울을 재야 한다.

영희 아이디어

동전을 2개씩 짝을 지어, n/2 짝을 각각 저울에 달아서 가짜 동전을 찾아보자.



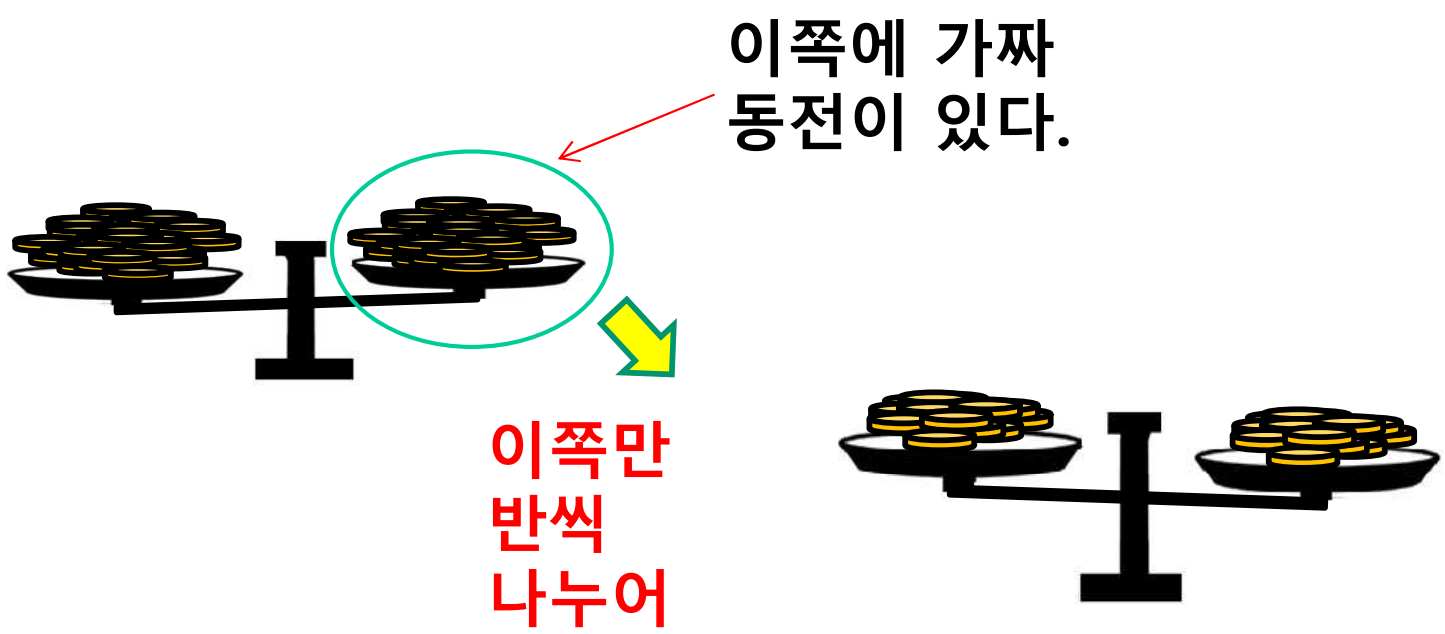


- 1 ~ n/2회

  최악의 경우 저울 사용 횟수가 거의 1/2로 감소

광수 아이디어

동전 더미를 반으로 나누어 저울 양편에 놓아보자.



- 남은 동전 수는 계속 1/2로 줄어들고, 나중에 2개가 남았을 때 가짜 동전을 가려낸다.

알고리즘

- 동전 더미를 반으로 나누어 저울에 달고, 가벼운 쪽의 더미를 계속 반으로 나누어 저울에 단다.
- 분할된 더미의 동전 수가 1개씩이면 마지막으로 저울을 달아 가벼운 쪽의 동전이 가짜이다.
- 이 알고리즘은 운이 좋을 때가 없다.

  왜냐하면 마지막에 가짜 동전을 찾기 때문이다.

1,024개 있을 때 몇 번 저울에 달아야 할까?

먼저 512개씩 양쪽에 올려놓고 저울을 재고,

다음은 256개씩 재고,

128개씩, 64개씩, 32개씩, 16개씩, 8개씩, 4개씩, 2개씩,

마지막엔 1개씩 올려서 저울을 잰다.

이는 총 10번이고, $\log_2 1{,}024 = 10$ 이다.

- n개 있다면 $\log_2 n$번 저울에 달면 가짜 동전을 찾는다.

최악의 경우 비교

철수는 1,023번 저울에 달아야 한다.

영희는 512번 달아야 한다.

광수는 10번이면 가짜 동전을 찾는다.

- 이 차이는 n이 커지면 더욱 더 커진다.

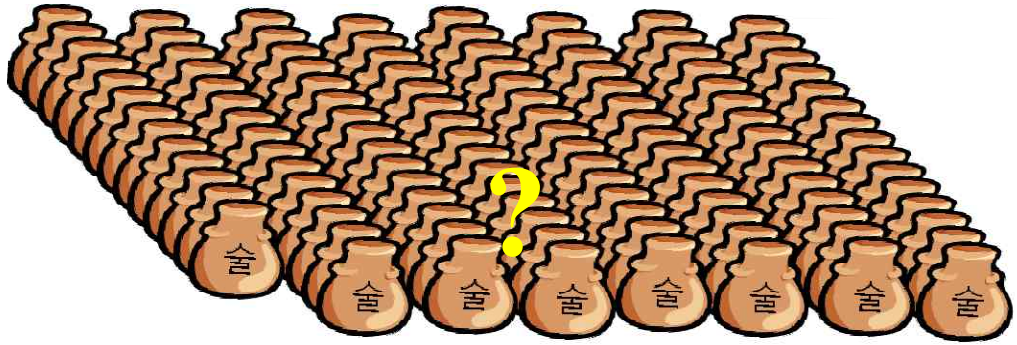
문제

1.7 독이 든 술단지

- 옛날 어느 먼 나라에 술을 매우 즐겨 마시는 임금님이 살고 있었다.

  창고에는 많은 술단지가 보관되어 있었다.

  어느 날 이웃 나라의 스파이가 창고에 들어가서 술단지 하나에 독을 넣고 나오다가 붙잡혔다.



- 스파이는 눈으로 확인할 수 없는 독을 사용하였고, 어떤 단지인지는 모르지만 하나의 단지에만 독을 넣었다고 말하고 죽었다.
- 스파이가 사용한 독의 특징

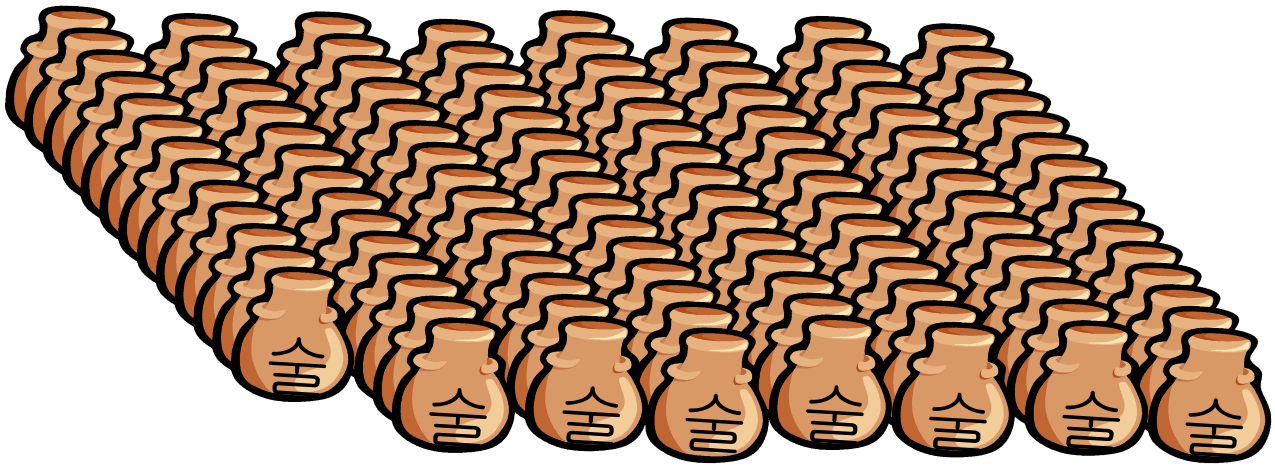
  독이 든 단지의 술을 아주 조금만 맛보아도 정확히 일주일 후에 죽는다.

문제

➢ 임금님은 신하들에게 명하였다.

독이 든 술단지를 반드시 일주일 만에 찾아내라.

단, 희생되는 신하의 수를 가능한한 줄여야 한다.



어떻게 독이 든 술단지를 일주일 만에 찾을 수 있을까?

- 이러한 문제 해결의 핵심은 적은 수의 술단지에 대해서 생각해 보자.

  즉, 술단지의 수가 2개일 때, 4개일 때 문제를 풀어 보고,

  술단지의 수를 늘려가면서 일반적인 규칙을 찾아보자.

입력의 크기가 아주 작을 때에 문제의 답을 찾아서 힌트를 얻어보자.

술단지가 2개일 때

- 한 명의 신하가 1개의 술단지의 술을 맛본다.

- 일주일 후

  신하가 살아 있으면, 맛보지 않은 단지에 독이 있다.

  신하가 죽는다면, 맛본 술단지에 독이 들어 있다.

술단지가 4개일 때

- 세 사람이 각각 한 개씩 맛본다면

- 두 사람으로 줄일 수는 없을까?

술단지가 4개일 때

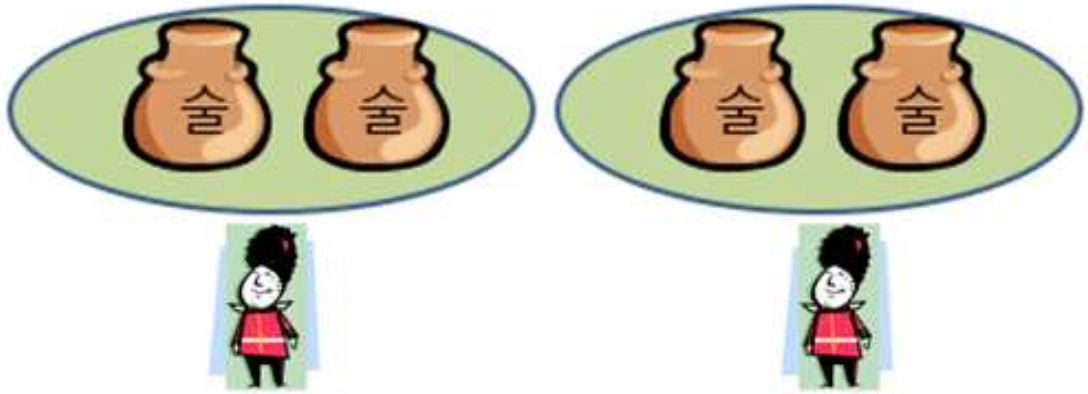
두 사람이 각각 한 단지씩 맛보게 해보면,

➢ 문제점

맛보지 않은 나머지 2개의 단지 중 하나에 독이 들어 있으면, 일주일 후 두 사람은 살아있을 것이고, 나머지 2개의 단지 중 어느 하나에 독이 들어 있는지를 알 수 없다.

술단지가 4개일 때

4개의 단지를 두 그룹으로 나누어, 각 그룹에 한 사람씩 할당해보자.



이는 맨 처음 생각해봤던 술단지의 수가 2인 경우가 2개가 생긴 셈이다.

과연 답을 찾을 수 있을까?

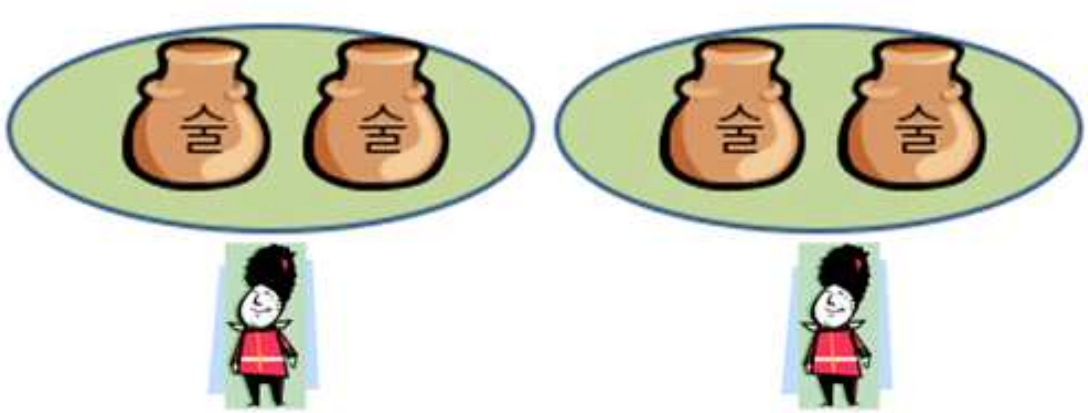
술단지가 4개일 때

- 문제점

일주일 후에 만일 두 사람 다 살아 있으면, 어떤 단지에 독이 들어 있나?

각 그룹에서 맛을 안 본 단지가 하나씩 있으므로, 이 2개의 단지 중 하나에 독이 들어 있는 것이다.

그러나 어느 단지인지를 알려면 또 일주일이 필요하므로 이 제안 역시 실패!



문제를 다시 살펴보자!

- 문제에서 간과한 점이 있나, 아니면 문제에 없는 조건을 우리 마음대로 만들었는지를 생각해 보자.
- 한 명의 신하가 반드시 하나의 단지의 술만 맛보아야 하는 것으로 생각했다.
- 문제에는 이러한 조건이 없다.

  이것이 문제 해결의 실마리이다.

아이디어

- 4개의 단지 중에서, 시음하지 않은 두개의 단지가 있다.

이 두 개의 단지 중에 하나를 두 신하에게 동시에 맛을 보게 하자.

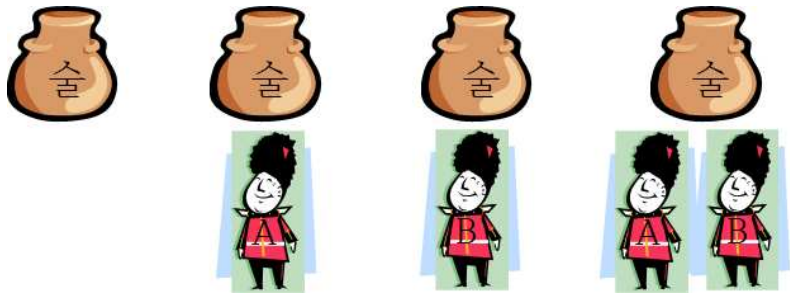
- 이렇게 하면 4가지의 결과가 생긴다.

두 신하를 각각 A와 B라고 하면

4가지 경우

➢ 두 신하를 각각 A와 B라고 하면

1. 아무도 시음하지 않은 단지에 독이 있으면,
   일주일 후에 두 신하 둘 다 살아있다.
2. A가 혼자 시음한 단지에 독이 있으면,
   일주일 후에 A만 죽는다.
3. B가 혼자 시음한 단지에 독이 있으면,
   일주일 후에 B만 죽는다.
4. A와 B 둘 다 시음한 단지에 독이 있으면,
   일주일 후에 둘 다 죽는다.



알고리즘

각 단지에 2진수를 0부터 부여하고,

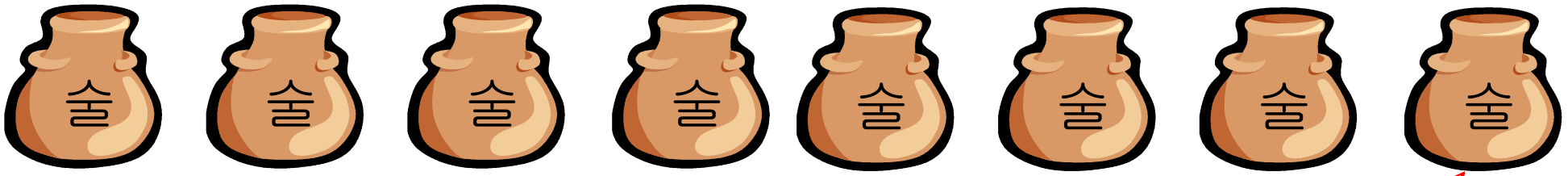
- 첫 비트가 1이면 신하 1이 맛본다.
- 둘째 비트가 1이면 신하 2가 맛본다.

…

- K번째 비트가 1이면 신하 K가 맛본다.

술단지가 8개일 때

0	0	0	0	0	1	0	1	0	0	1	1	1	0	0	1	0	1	1	1	0	1	1	1
					A		B			B	A	C			C		A	C	B		C	B	A



- 각 신하는 4개의 술단지를 맛본다.
- 첫 비트가 1이면 A가 맛본다.
- 둘째 비트가 1이면 B가 맛본다.
- 셋째 비트가 1이면 C가 맛본다.

여기에 독이 있으면 3명 모두 죽는다.

술단지 수가 n이면

- 희생자 수 = 0 ~ $\log_2 n$

요약

- 순차 탐색(Sequential Search): 주어진 순서에 따라 차례로 탐색
- 이진 탐색(Binary Search): 정렬된 데이터에 대해서 중간에 있는 데이터를 비교하여 그 결과에 따라 같으면 탐색을 마치고, 다르면 작은 쪽 또는 큰 쪽을 같은 방식으로 탐색
- 동전 거스름돈 문제에서 가장 액면이 높은 동전을 항상 욕심 내어 선택한다. 그리디 (Greedy) 알고리즘(4장)
- 한붓그리기 문제는 오일러 서킷(Euler Circuit) 문제와 같다. 알고리즘의 핵심은 현재 점에서 사이클이 존재하면 진행한다.

요약

- 가짜 동전 찾기에서 동전더미를 반으로 분할하여 저울에 달고, 가짜 동전이 있는 더미를 계속해서 반으로 나누어 저울에 단다. 분할 정복 (Divide-and-Conquer) 알고리즘(3장)
- 독이든 술 단지 문제는 2진수를 활용하여 해를 찾는다.