

소프트웨어 프로젝트 1

계산적 사고와 알고리즘

2022학년도 1학기

국민대학교 소프트웨어학부

제 1 부

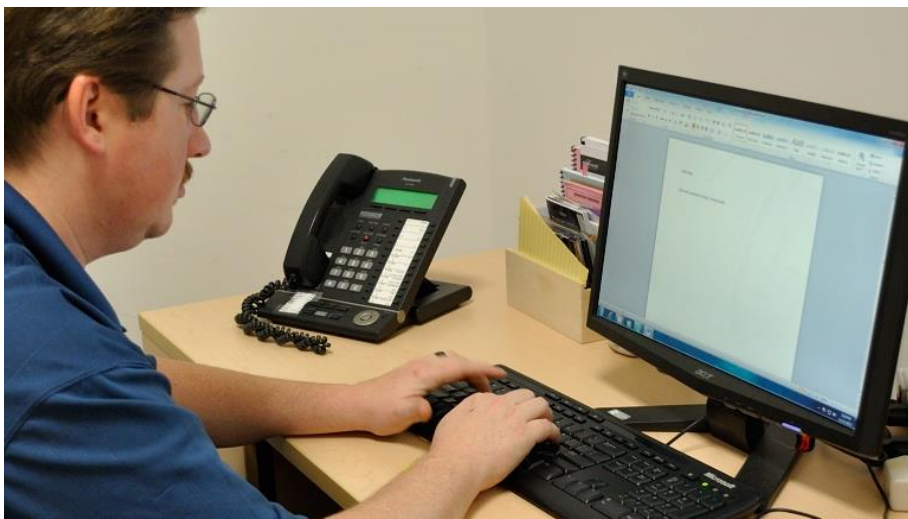
전공으로서의 소프트웨어

컴퓨터?



특정한 곳에서 특정한 사람들에 의해 이용되던
"숨어있는" 기계

누구나 주머니 안에 하나씩 가지고 다니는
"사람에 붙어 다니는" 도구
(더 숨어서 아마도 모습을 감출 듯)



Computers are everywhere!

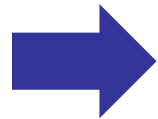
컴퓨터가 "모든 측면에서" 사람들의 삶을 달라지게 하고 있음

요즘 어린 아이들은 그게 달라지고 있는 줄도 모르고 있음



패러다임 (Paradigm)

어떤 한 시대 사람들의
견해나 사고를 근본적으로 규정하고 있는
테두리로서의 인식의 체계



시대를 지배하는 사고 체계

시대가 우리에게 요구하는 인식의 틀

패러다임의 변화

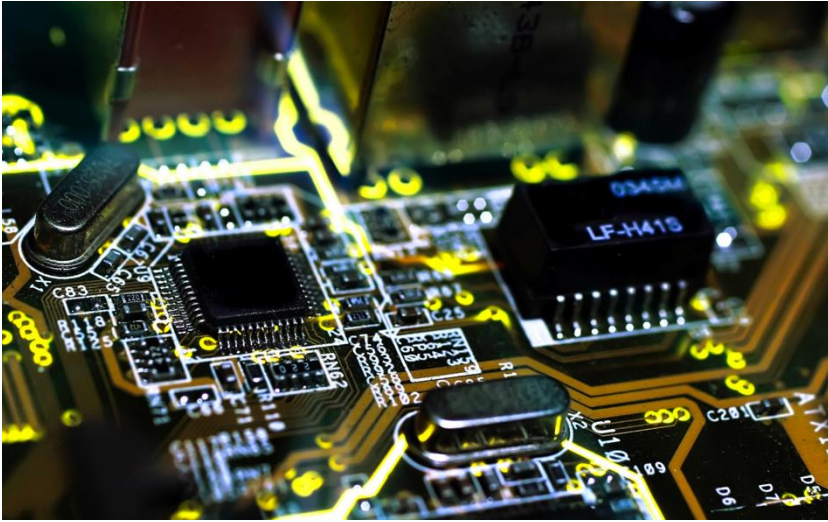
Intelligent Machines

Nvidia CEO: Software Is Eating the World, but AI Is Going to Eat Software

Jensen Huang predicts that health care and autos are going to be transformed by artificial intelligence.

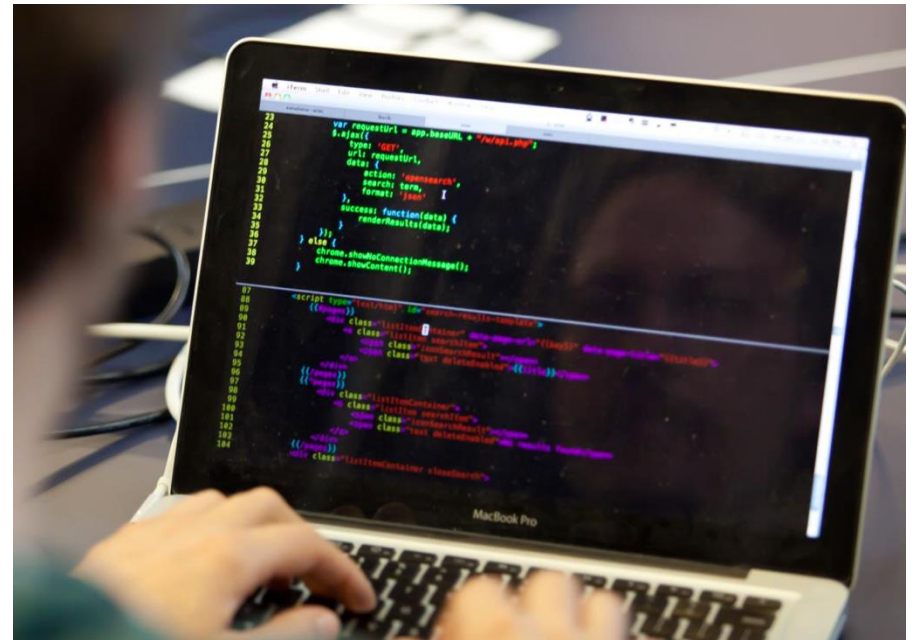


교과목으로서의 컴퓨터



하드웨어
(Hardware)

소프트웨어
(Software)



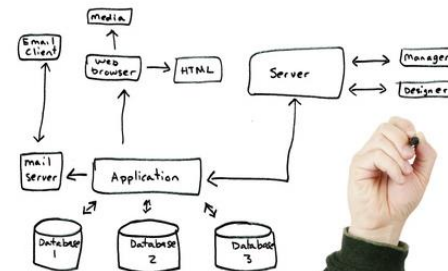
국어, 영어, 수학, 과학, 사회, 음악, 미술, 체육, ..., 컴퓨터?

도구로서의 컴퓨터, 소프트웨어

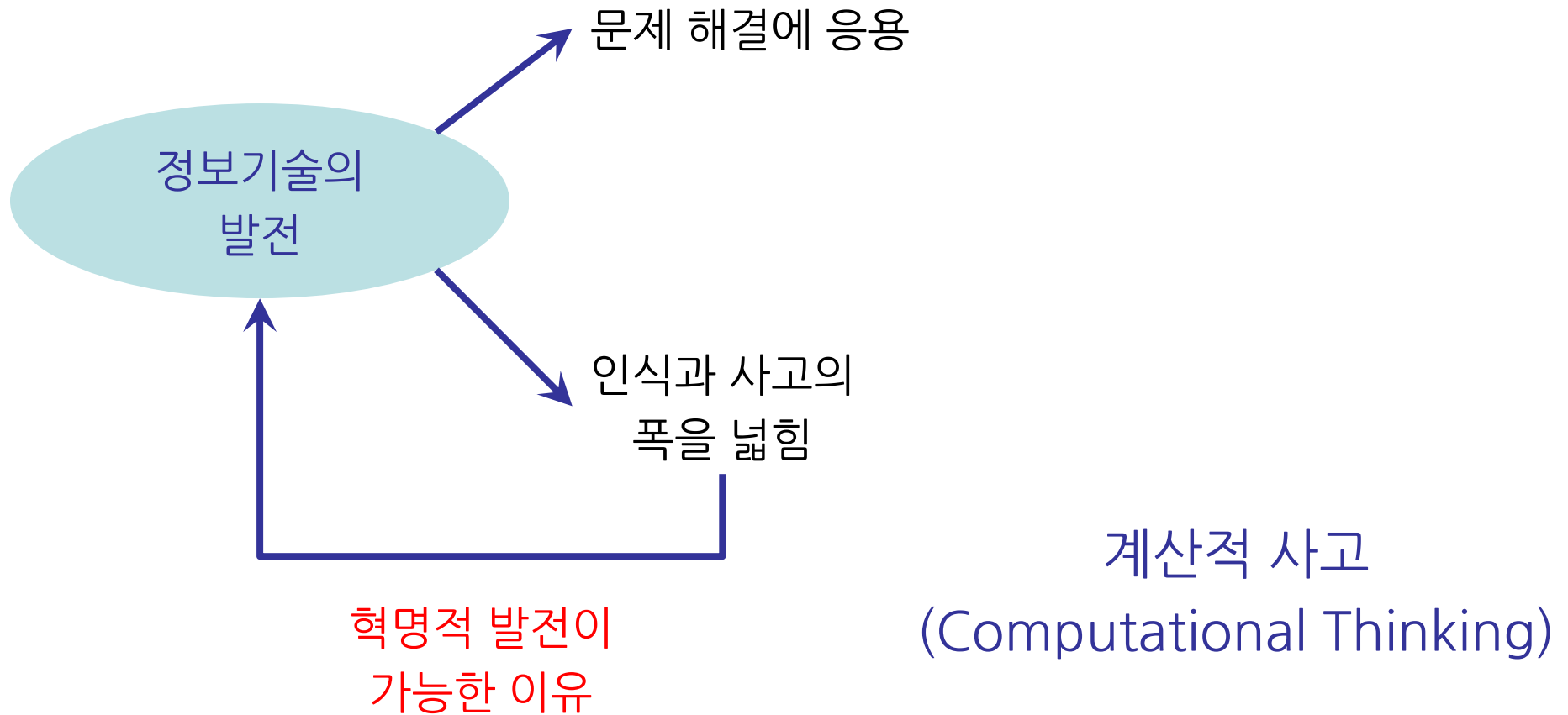
정보기술의
발전

문제 해결에 응용

인식과 사고의
폭을 넓힘

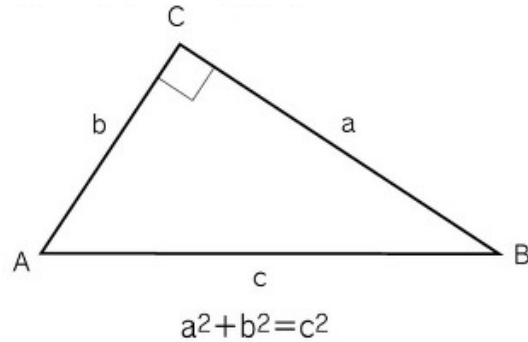
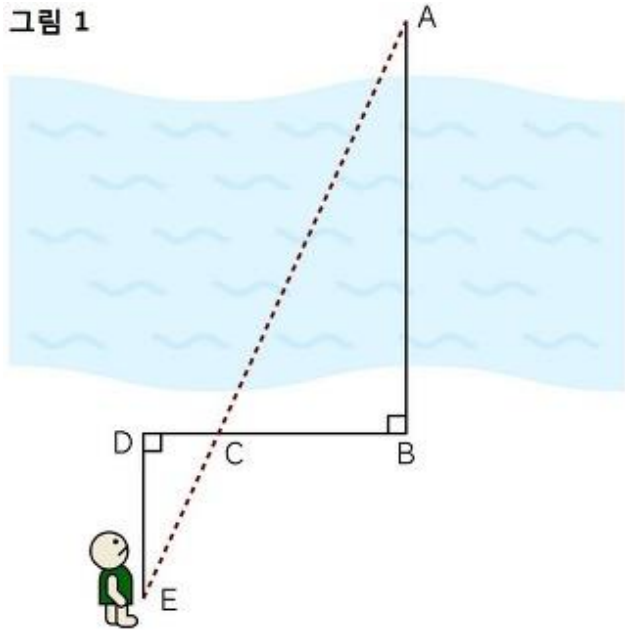


도구로서의 컴퓨터, 소프트웨어

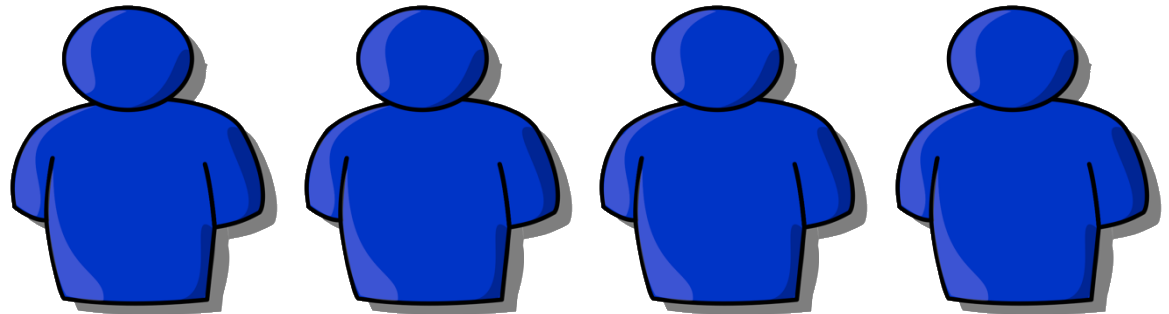


프로그래밍을 통한 사고의 훈련

그림 1



수학을 배우는 것과 비슷하다!



제 2 부

계산적 사고

(Computational Thinking)

계산적 사고

수업 태도가 매우 불량하구나.

그에 대한 벌로,

1 부터 100 까지를 모두 더해라!



Carl Friedrich Gauss (독일)

$$\begin{array}{r}
 1 + 2 + 3 + \cdots + 99 + 100 \\
 +) 100 + 99 + 98 + \cdots + 2 + 1 \\
 \hline
 101 + 101 + 101 + \cdots + 101 + 101 \\
 \\
 = 101 \times 100
 \end{array}$$

답은?

$$101 \times 100 / 2 = 5050$$

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

549 를 찾아라!

297	46	337	865	589	609	786	143	55	691
867	649	87	666	63	699	95	738	513	12
75	344	151	279	42	946	517	172	623	222
730	228	823	913	364	625	80	781	136	630
458	426	661	600	816	342	114	749	233	849
646	341	685	287	607	963	328	449	902	339
878	672	712	159	535	514	263	334	176	722
503	950	855	402	996	572	37	995	509	372
242	508	274	987	234	248	831	549	390	650
322	400	93	933	952	43	782	939	839	196

월리를 찾아라!



549 를 찾아라!

297	46	337	865	589	609	786	143	55	691
867	649	87	666	63	699	95	738	513	12
75	344	151	279	42	946	517	172	623	222
730	228	823	913	364	625	80	781	136	630
458	426	661	600	816	342	114	749	233	849
646	341	685	287	607	963	328	449	902	339
878	672	712	159	535	514	263	334	176	722
503	950	855	402	996	572	37	995	509	372
242	508	274	987	234	248	831	549	390	650
322	400	93	933	952	43	782	939	839	196

월리를 찾아라!



549 를 찾아라!

12	37	42	43	46	55	63	75	80	87
93	95	114	136	143	151	159	172	176	196
222	228	233	234	242	248	263	274	279	287
297	322	328	334	337	339	341	342	344	364
372	390	400	402	426	449	458	503	508	509
513	514	517	535	549	572	589	600	607	609
623	625	630	646	649	650	661	666	672	685
691	699	712	722	730	738	749	781	782	786
816	823	831	839	849	855	865	867	878	902
913	933	939	946	950	952	963	987	995	996



549 를 찾아라!

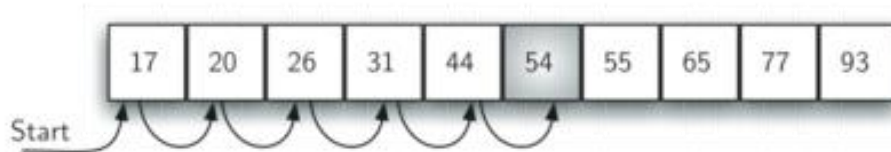
12	37	42	43	46	55	63	75	80	87
93	95	114	136	143	151	159	172	176	196
222	228	233	234	242	248	263	274	279	287
297	322	328	334	337	339	341	342	344	364
372	390	400	402	426	449	458	503	508	509
513	514	517	535	549	572	589	600	607	609
623	625	630	646	649	650	661	666	672	685
691	699	712	722	730	738	749	781	782	786
816	823	831	839	849	855	865	867	878	902
913	933	939	946	950	952	963	987	995	996



컴퓨터에게 시킨다면?

알고리즘적 사고

작은 것이 앞에 있고 큰 것이 뒤에 있으니,
앞에서부터 시작해서 549 를 만날 때까지 찾아가라!

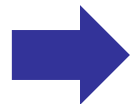


어떤 경우 금방 찾나요?

어떤 경우 찾는 데 오래 걸리나요?

오래 걸리면, 얼마나 걸리나요?

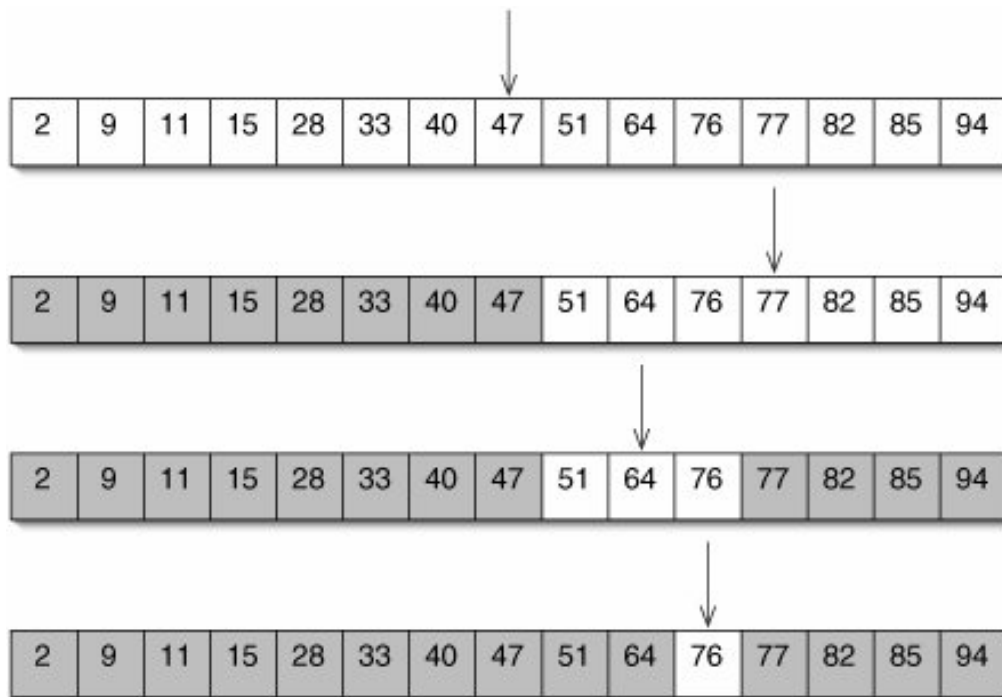
평균적으로는 얼마나 걸리나요?



평균적으로 더욱 빠르게 찾는 방법은 없을까?

보다 나은 문제의 해결법

반으로 잘라서, 없는 것이 확실한 쪽은 버린다!



우선은 순서대로 정렬해야

54	26	93	17	77	31	44	55	20
----	----	----	----	----	----	----	----	----

Assume 54 is a sorted
list of 1 item

26	54	93	17	77	31	44	55	20
----	----	----	----	----	----	----	----	----

inserted 26

26	54	93	17	77	31	44	55	20
----	----	----	----	----	----	----	----	----

inserted 93

17	26	54	93	77	31	44	55	20
----	----	----	----	----	----	----	----	----

inserted 17

17	26	54	77	93	31	44	55	20
----	----	----	----	----	----	----	----	----

inserted 77

17	26	31	54	77	93	44	55	20
----	----	----	----	----	----	----	----	----

inserted 31

17	26	31	44	54	77	93	55	20
----	----	----	----	----	----	----	----	----

inserted 44

17	26	31	44	54	55	77	93	20
----	----	----	----	----	----	----	----	----

inserted 55

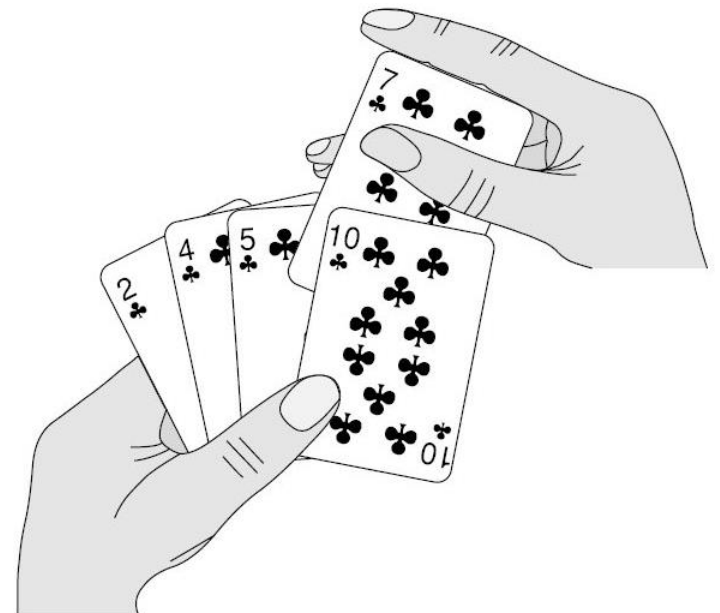
17	20	26	31	44	54	55	77	93
----	----	----	----	----	----	----	----	----

inserted 20

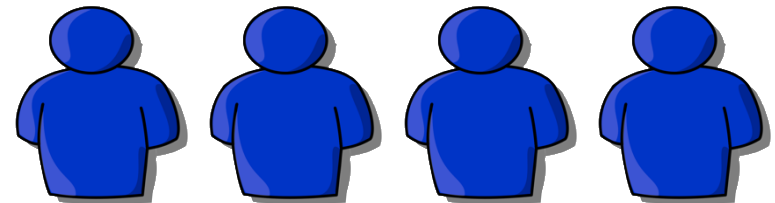
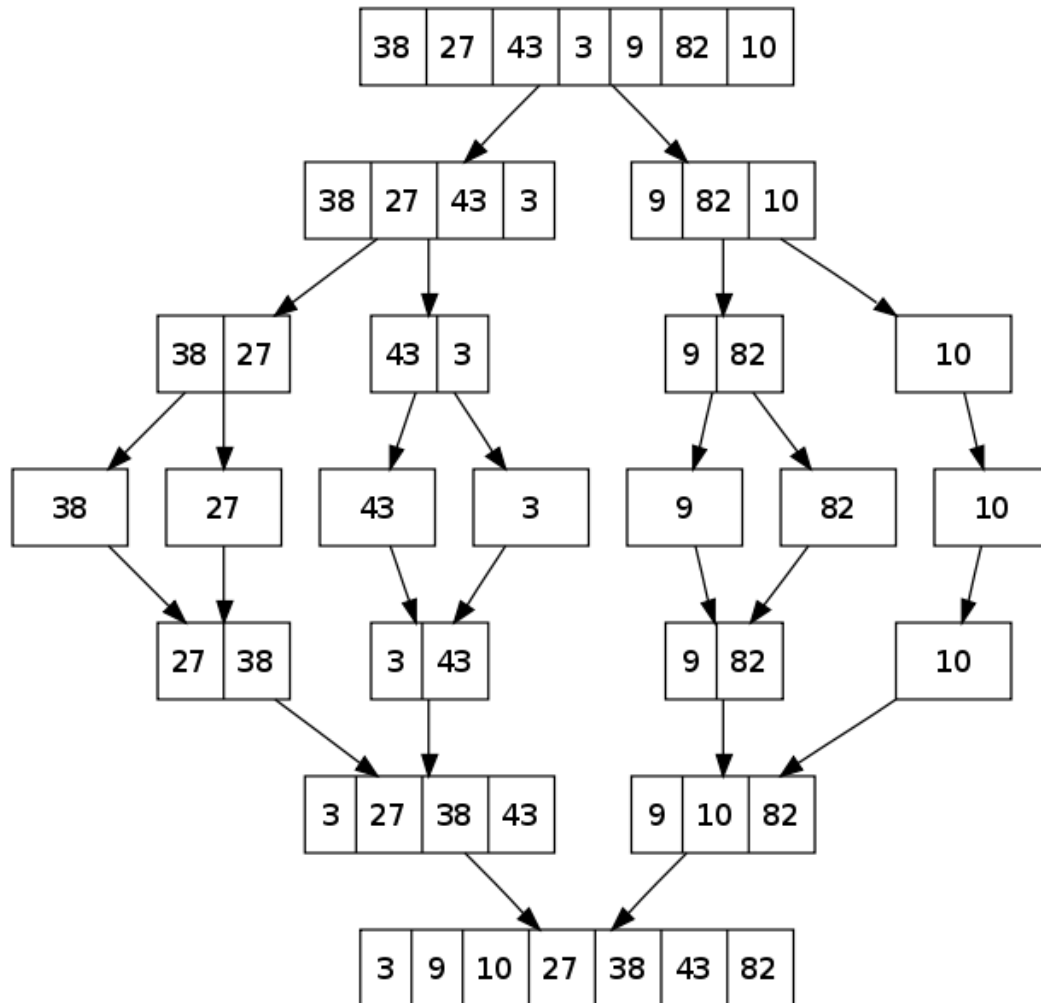
어떤 경우 금방 정렬하나요?

어떤 경우 오래 걸리나요?

평균적으로는 얼마나 걸리나요?



보다 빨리 정렬하는 방법?



관측

1. 삶은 문제와 해결의 연속이다.
2. 문제의 해결 방식은 한 가지만 있는 것이 아니다.
3. 말로만 하면 쉽다. → 프로그래밍도 쉬울까?
4. 환경과 도구는 어느 때보다도 빠르게 변화한다.

프로그래밍 - 문제의 해결

- 사고적 측면 (Thinking - Design)
- 기술적 측면 (Skill - Implementation)

알고리즘이란?

Finite set of instructions that accomplish a particular task

→ 특정한 작업을 달성하기 위한 명령들의 유한한 집합

알고리즘이 만족해야 할 조건들:

- Input - 입력은 외부로부터 주어질 것
- Output - 적어도 하나 이상의 해답을 출력할 것
- Definiteness - 명료하고 모호성이 없어야 함
- Finiteness - 유한한 수의 단계를 거친 후 반드시 종료해야 함
- Effectiveness - 모든 명령은 기본적인 것으로 이루어져야 함 (연필과 종이로...)
- Feasible - 주어진 문제에 대한 답을 **웬만한** 시간 내에 찾아내야 함

예: 최댓값 찾기

54	32	86	19	73	27	48	63
----	----	----	----	----	----	----	----

맨 앞에서 시작하면서 수를 하나 하나 꺼내서,

지금까지 등장한 수들 중 가장 컸던 것과 비교하여

(크면) 갱신

(아니면) 그냥 유지

- 그렇다면 “지금까지 등장한 수들 중 가장 컸던 것”에 대하여

(1) 기억 (기록) 을 유지할 수 있어야 하고

(2) 처음에는 어떻게 할지를 결정해 두어야 함

예: 최댓값 찾기

54	32	86	19	73	27	48	63
----	----	----	----	----	----	----	----

앞에서 설명한 알고리즘은,

- (1) 얼마의 시간을 소요할까?
- (2) 수가 많아지면 소요 시간이 늘어날까, 줄어들까, 이도 저도 아닐까?
- (3) 수가 많아지면 (입력의 크기가 커지면) 소요 시간은 무엇에 비례할까?

예: 최댓값 찾기

54	32	86	19	73	27	48	63
----	----	----	----	----	----	----	----

앞에서 설명한 알고리즘은,

- (1) 얼마의 시간을 소요할까?
- (2) 수가 많아지면 소요 시간이 늘어날까, 줄어들까, 이도 저도 아닐까?
- (3) 수가 많아지면 (입력의 크기가 커지면) 소요 시간은 무엇에 비례할까?

$$O(n)$$

다른 예: 최대값 찾기를 이용한 정렬

54	32	86	19	73	27	48	63
----	----	----	----	----	----	----	----

모든 수들 중 가장 큰 것을 하나 고르고 (앞의 알고리즘 적용)

나머지 수들에 대해서 이 과정을 반복하면서

각 단계에서 골라진 수들을 그 순서대로 나열하면

→ 이것이 이 수들을 크기 순서대로 (내림차순) 정렬한 것

다른 예: 최댓값 찾기를 이용한 정렬

54	32	86	19	73	27	48	63
----	----	----	----	----	----	----	----

86	54	32	19	73	27	48	63
----	----	----	----	----	----	----	----

86	73	54	32	19	27	48	63
----	----	----	----	----	----	----	----

86	73	63	54	32	19	27	48
----	----	----	----	----	----	----	----

다른 예: 최댓값 찾기를 이용한 정렬

86	73	63	54
----	----	----	----

32	19	27	48
----	----	----	----

86	73	63	54	48
----	----	----	----	----

32	19	27
----	----	----

86	73	63	54	48	32
----	----	----	----	----	----

19	27
----	----

86	73	63	54	48	32	27
----	----	----	----	----	----	----

19

다른 예: 최댓값 찾기를 이용한 정렬

86	73	63	54	48	32	27	19
----	----	----	----	----	----	----	----

이 결과를 얻기까지,

몇 번이나 “최댓값 고르기” 를 했나요?

각 단계의 “최댓값 고르기” 에서는 몇 개의 수를 꺼내어 비교했나요?

다른 예: 최댓값 찾기를 이용한 정렬

86	73	63	54	48	32	27	19
----	----	----	----	----	----	----	----

이 결과를 얻기까지,

몇 번이나 “최댓값 고르기” 를 했나요?

각 단계의 “최댓값 고르기” 에서는 몇 개의 수를 꺼내어 비교했나요?

$$8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 36$$

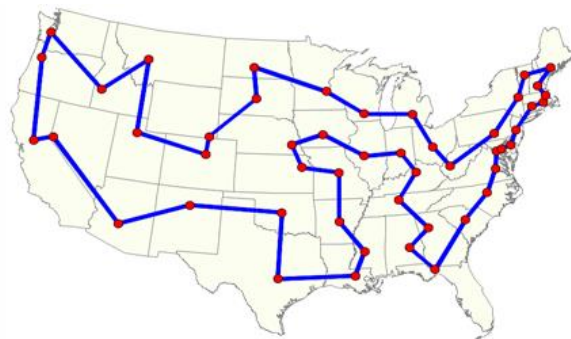
$$n + (n - 1) + (n - 2) + \dots + 1 = n(n + 1) / 2$$

$$O(n^2)$$

또 다른 예: 외판원 문제

(Traveling Salesperson Problem; TSP)

여러 도시들이 있고 한 도시에서 다른 도시로 이동하는 비용이 모두 주어졌을 때,
모든 도시들을 단 한 번만 방문하고 원래 시작점으로 돌아오는
최소 비용의 이동 순서를 구하시오.



이것은 “매우” 어려운 문제입니다.

- 매우? 얼마나?
- “어렵다” 는 것은 무슨 뜻이길래?

컴퓨터과학에서는

이런 것들을 흔히 다룹니다.

Q & A