

Pre-school

개발자에게 자료구조란?

<넥슨 박종천 부본부장님 강연 중>

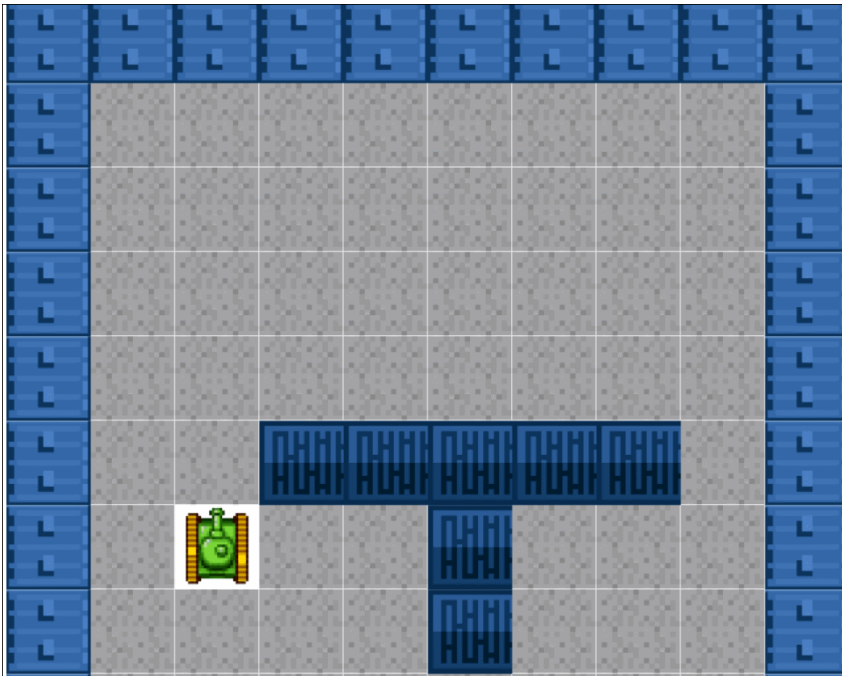
Foundation Knowledge Quiz

- Recursion, Index, Sort, Binary Search
- Stack / Heap
- Process / Thread
- Call by Value / Call by Reference
- Native Code
- MVC Architecture
- Write simple text-based games



여러분은 이 퀴즈 중 얼마나 많이 알고 있습니까?

자료구조 왜 배우나요?



가장 기초적인 AI의 일종
- A star 알고리즘

구현하는 데 필요한 것

1. Priority Queue
2. Stack(or linked List)
3. 피타고라스 정리(?)
4. 대소비교(?)

자료구조만 알고 있으면
AI를 구현할 수 있다고??

```
|class Pathfinder
{
public:
    PriorityQueue<PathBlock*> * pQueue; //open node 담는다.
    PathBlock * pPathBlocks[10][10]; //F, G, H 정보와 상태를 담는다.
    Stack<Position> * pList;
```

자료구조 왜 배우나요?



Taehwan Yang

7월 21일 · 🌐 ▼

Linus Tovalds said,
"I can't do UI to save my life!

If i was stranded on an island and the only way to get off the island was to make the pretty UI.... 더 보기



리눅스의 기본 철학

리누스 토발즈는 두 번의 기술 변혁을 이뤄냈습니다. 첫 번째는 리눅스 커널로, 이것으로 인터넷이 작동하게 되었습니다. 두 번째는 전세계 개발자들이 사용하는 소스코드 관리 시스템인 깃입니다. TED 큐레이터인 크리스 앤더슨과의 인터뷰에서 토발즈는...

타임라인
14:27~15:55

리누스 토발즈가
리눅스에 대해 이야기 하면서
링크드 리스트에 대해
이야기 합니다.

개발자라면 TED에서 개발
관련 영상을 볼 수 있어야 하지
않을까요?

자료구조와 알고리즘

많은 분들이 자료구조와 알고리즘을 혼동해 사용하거나 같은 거라고 생각합니다. 하지만 이 두 가지는 다릅니다. 물론 밀접하게 연관되어 있지요.

1. 자료구조

: 데이터를 어떤 구조로 저장하고 탐색해야 가장 효율적인가?

2. 알고리즘

: 문제를 해결하는 방법론

자료구조와 알고리즘

1. 자료구조의 알고리즘
: 데이터를 저장하고 탐색하는 방법에 대한 고민들
2. 자료구조를 이용한 알고리즘
: 자료구조를 이용해 어떤 문제를 해결하는 것

자료구조의 구성

1. Insert

: 데이터를 어떻게 저장할 것인가?

2. Search

: 데이터를 어떻게 탐색할 것인가?

3. Delete

: 필요 없어진 데이터를 어떻게 지울 것인가?

재귀함수

재귀함수란?

```
long long SumToNumber(int num)
{
    //탈출조건
    if (num == 1) 탈출조건
        return 1;

    return SumToNumber(num - 1) + num;
}
```

같은 이름의 함수가 함수 내에 나올 때 이를 재귀함수라 부릅니다.

재귀함수

실습 예제 : 1부터 1억까지의 수를 모두 더하는 프로그램

세 가지 알고리즘으로 설계를 해봅시다!

```
//1. recursion 재귀함수
```

```
sum = SumToNumber(num);
```

```
//2. for 문 for 문
```

```
for (int i = 0; i <= num; i++)  
    sum += i; 가장 일반적으로 떠올릴 알고리즘
```

```
//3. 등차수열 등차수열
```

```
sum = ArithmeticSequence(num);
```

재귀함수

1. 재귀함수를 이용한 방법

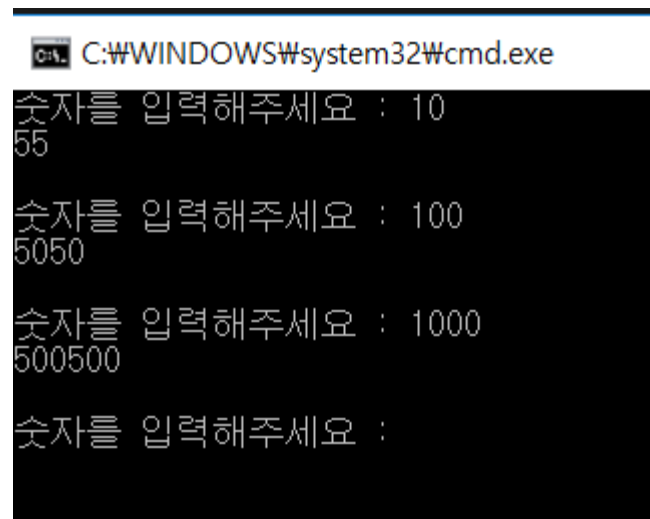
```
long long SumToNumber(int num)
{
    //탈출조건
    if (num == 1)
        return 1;

    return SumToNumber(num - 1) + num;
}
```

재귀함수

1. 재귀함수를 이용한 방법

10, 100, 1000까지는 잘 돌아갑니다



```
C:\WINDOWS\system32\cmd.exe
숫자를 입력해주세요 : 10
55
숫자를 입력해주세요 : 100
5050
숫자를 입력해주세요 : 1000
500500
숫자를 입력해주세요 :
```

여기에 100,000,000 을 입력해 봅시다.

재귀함수

Stack overflow

Microsoft Visual Studio



Unhandled exception at 0x00CD1739 in Recursion.exe: 0xC00000FD: Stack overflow (parameters: 0x00000001, 0x00CE2F74).

Stack overflow, 즉 스택이 터졌습니다!!

☐ Break when this exception type is thrown

[Break and open Exception Settings](#)

Break

Continue

Ignore

재귀함수

for 문을 이용한 방법 vs 등차 수열을 이용한 방법

두 가지 알고리즘 모두 잘 돌아갑니다

```
//2. for 문  
for (int i = 0; i <= num; i++)  
    sum += i;  
  
//3. 등차수열  
sum = ArithmeticSequence(num);
```

성능의 측면에서는 어떨까요?

재귀함수

for 문을 이용한 방법

10,000,000 을 입력하면 1억 번 돌아야 겠네요.....

```
for (int i = 0; i <= num; i++)  
    sum += i;
```

재귀함수

등차 수열을 이용한 방법

$$1 + 2 + + num$$

등차 수열 공식

n : 항의 개수
a : 첫째 항
l : 마지막 항

$$\frac{n(a + l)}{2} = \frac{(num) * (1 + num)}{2}$$

```
long long ArithmeticSequence(long long num)
{
    long long sum = (num * (1 + num)) / 2;
    return sum;
}
```

수식 한번 계산 하면 끝이네요!!!!

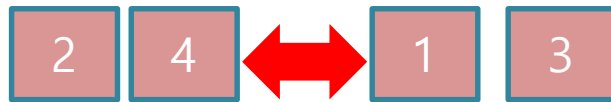
정렬 알고리즘

버블 정렬

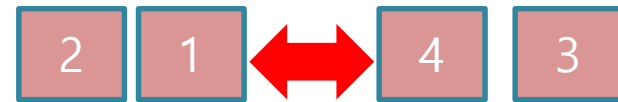


비교

4가 더 크므로 바뀌지 않는다



비교



교환



비교



교환

비교 3회

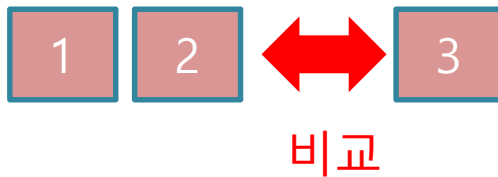


가장 큰 수가 맨 뒤로 가 있다!!

정렬 알고리즘

버블 정렬

이미 정렬된 4를 빼고 나머지에서
같은 알고리즘을 적용



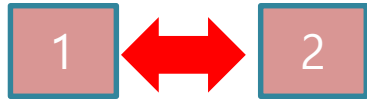
비교 2회

가장 큰 수가 맨 뒤로 가 있다!!

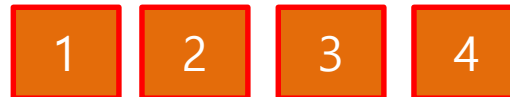
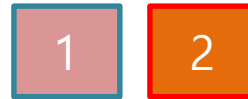
정렬 알고리즘

버블 정렬

이미 정렬된 3, 4를 빼고 나머지에서
같은 알고리즘을 적용



비교



비교 1회

정렬이 완료됨

정렬 알고리즘

버블 정렬

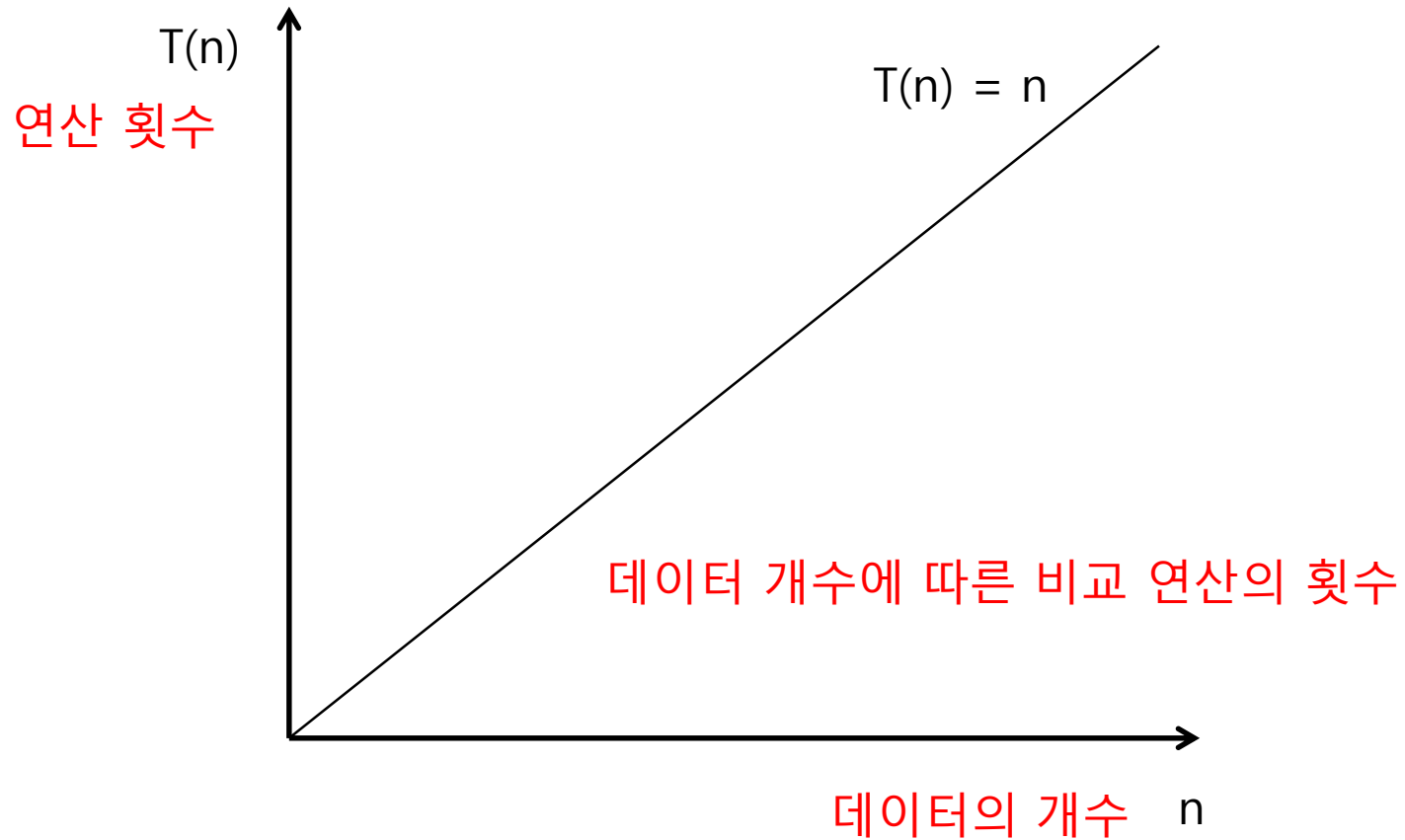
배열 길이가 4라면

```
void BubbleSort(int arr[], int len)
{
    i는 0부터 2
    for (int i = 0; i < len - 1; i++)
    {
        for (int j = 0; j < len - i - 1; j++)
        {
            if (arr[j] > arr[j + 1])
                Swap(&arr[j], &arr[j + 1]);
        }
        i == 0 → j == 0, 1, 2 : 3회 비교
        i == 1 → j == 0, 1 : 2회 비교
        i == 2 → j == 0 : 1회 비교
    }
}
```

정렬 알고리즘

버블 정렬

시간복잡도(Time Complexity)



정렬 알고리즘

버블 정렬

정렬이 끝날 때까지 몇 번 비교하는가?

```
void BubbleSort(int arr[], int len)
{
    for (int i = 0; i < len - 1; i++)
    {
        for (int j = 0; j < len - i - 1; j++)
        {
            if (arr[j] > arr[j + 1]) 비교연산
                Swap(&arr[j], &arr[j + 1]);
        }
    }
}
```

정렬 알고리즘

버블 정렬

데이터 개수가 4개 일 때

3번 비교 → 2번 비교 → 1번 비교

$$1 + 2 + \dots + n-1$$

등차수열 공식

n : 항의 개수
a : 첫째 항
l : 마지막 항

$$\frac{n(a + l)}{2} = \frac{(n - 1) * (1 + n - 1)}{2} = \frac{n - 1 + 1 + 2n + n^2}{2}$$

빅오 계산할 때는 계수 상수 다 떼고 가장 큰 수만 남긴다.

$O(n^2)$ 데이터가 많아질수록 기하급수적!!

감사합니다.