

2. Non-Linear Structure, Optimization, and Algorithms



- Genetic Algorithm
- Goal
- Difficult Problem
- Optimization

Application 2: Genetic Algorithm

Genetic : 유전

상당히 오래됨

최근에 개발된 시스템에서 구현은 안함

옛날 방식이지만 개념적으로 받아들이기에 좋음

Weekly Objectives

- ◆ This week, we study the genetic algorithm and the traveling salesman problem.
- ◆ Objectives are
 - ◆ Understanding when and why the genetic algorithm is used
 - ◆ Memorizing the structure of the genetic algorithm
 - ◆ Understanding the encoding strategies of the genetic algorithms
 - ◆ Understanding the roles and the rational of the algorithm steps
 - ◆ Selection step
 - ◆ Crossover step
 - ◆ Mutation step
 - ◆ Substitution step

언제 **genetic algorithm**이 쓰이는지
(문제에 맞는 프로그래밍 언어를 찾아서
적용하는것이 엔지니어의 능력)

A difficult problem...

Easy Problem



Finding a maximum value of a structure
 $O(\log N)$ or $O(N)$

Difficult Problem



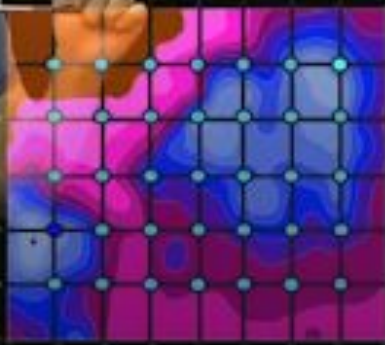
Finding a travel plan minimizing the trip time
 $O(N!)$

- What makes a problem difficult?
 - Difficulty in implementing an algorithm?
 - Difficulty in understanding an algorithm?
 - Difficulty in calculating a solution in a given time, or your lifetime...
- Imagine your algorithm solves a problem size X in a month...
 - with $O(N!)$ complexity
 - If you increase a size to $(X+1)$, then now it takes $(X+1)$ month to solve it

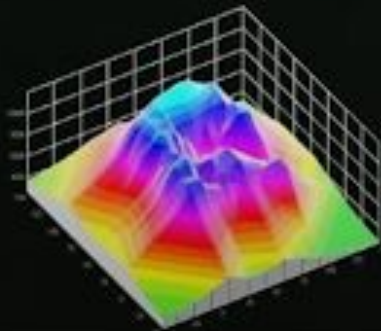
6개의 데이터로
1일이 걸리던 문제에 데이터가
하나만 늘어나도 7일이 되어버린다

Search space and global optimum

- ◆ Optimization is tightly linked to space search
 - ◆ Finding a maximum (a type of optimum) in a binary search tree
 - ◆ Search through a BST following a pre-defined paths
 - ◆ Data structure is a safe path with a consistency keeping the assumptions
- ◆ What-if there is no paths, just a big list sizing $N!$
 - ◆ Searching a record by a record must not be feasible
 - ◆ The list is not structured with a pattern
 - ◆ No deterministic ways to search the space!



Finding the
highest point
in the list



답이 될 수 있는 값은 아주 많은데
그중에 가장 좋은 답은 하나이다
라는게 optimization 을 search로
바라보는 사례