Motivations of Data Structures and Algorithms

Hsuan-Tien Lin

Dept. of CSIE, NTU

March 3, 2020

Introduction of Algorithms

What is Algorithm?

譜

樂譜

暗器譜

食譜

程式譜

descriptions to get something done correctly/efficiently by computer

Five Criteria of Algorithm

Knuth

"correct"

3 5 1 4 2

programmer terminate computer

input, output, definiteness, finiteness, effectiveness

getMinIndex with Sequential Search Algorithm

input

```
int getMinIndex(int* arr, int len){
  int minpos = 0; int i;
  for(i=1; i<len; i++){
    if (arr[i] < arr[minpos])
      minpos = i;
  }
  return minpos;
}
```

finiteness

effective (compiler)

definiteness (C lang)

output

returns index to minimum element within array

Correctness Proof of Algorithm

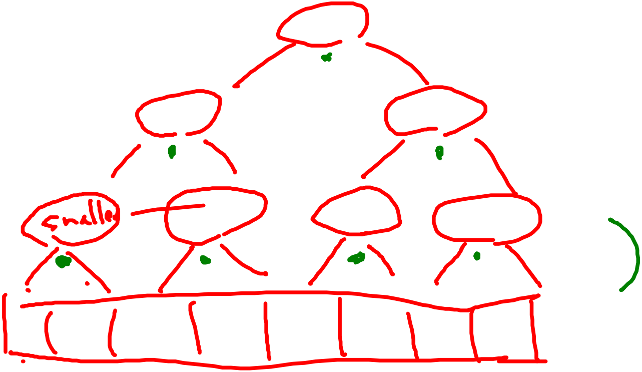
claim: "algorithm" returns m such that
arr[m] <= arr[j] for all j

claim2: at (end of) loop of i = k
arr[minpos] <= arr[j] for j = 0, 1, .., k

i=1: trivial
i=k ==> i=k+1

e.g. loop invariance by mathematical induction
—discrete math helps!

Efficiency of Algorithm



knockout tournament for `getMinIndex`: not much faster overall, but possibly faster if done in parallel

Expressing Algorithms with Pseudo Code

Pseudo Code for `getMinIndex`

口語

```
getMinIndex
    (integer array arr, integer len)
    minpos <- 0
    for i <- 1 to len do
      if arr[i] < arr[minpos] then
        minpos <- i
    return minpos
```

pseudo code: “spoken language” of programming

Bad Pseudo Code: Too Detailed

長舌

```
a = arr[i]
b = arr[minpos]
if a < b then ...
```

goal of pseudo code: communicate efficiently

Bad Pseudo Code: Too Mysterious

```
minpos, i

a = 0
for b = 1 to len-1
   if arr[b] < arr[a] then ...
```

goal of pseudo code: communicate correctly

Bad Pseudo Code: Too Abstract

惜字如金

run a loop that updates minpos in every iteration

goal of pseudo code: communicate effectively

Good Pseudo Code of SelSort

```
selSort
  (integer array arr, integer len)

  for i <- 0 to len-1 do
    // find minIndex from arr[i .. len-1]
    min <- getMinIndex(arr[i .. len-1])
    // put arr[min] at arr[i]
    swap(arr[min], arr[i])
```

no “formal definition” and depends on the speaker/listener

Introduction of Data Structures

What is Data Structure?

收纳方式

Cloth Structure

Food

Player

scheme of organizing data
within computer

How to Organize 200 Exam Sheets?

隨便

最高分 -〉 最低分

學號

依尾數分十份

different use cases

⟹ different organization scheme (data structure)

Good Algorithm Needs Proper Data Structure

if having data structure such that `getMinIndex` *faster*,

$\Longrightarrow$ `SelSort` *also faster* (we will see)

algorithm :: data structure $\sim$ recipe :: kitchen structure

Good Data Structure Needs Proper Accessing Algorithms: `get`, `insert`

rule of thumb for speed: often-`get` ⇔ “nearby”

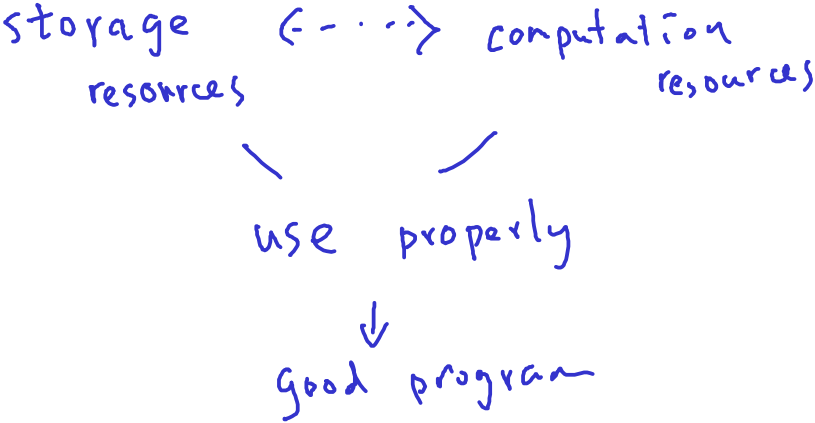
Good Data Structure Needs Proper Maintenance

Algorithms: `construct`, `update`, `remove`

hidden “cost” of data structure: maintenance effort

Why Data Structures and Algorithms?

Why Data Structures and Algorithms?



storage ⟷ computation

resources resources

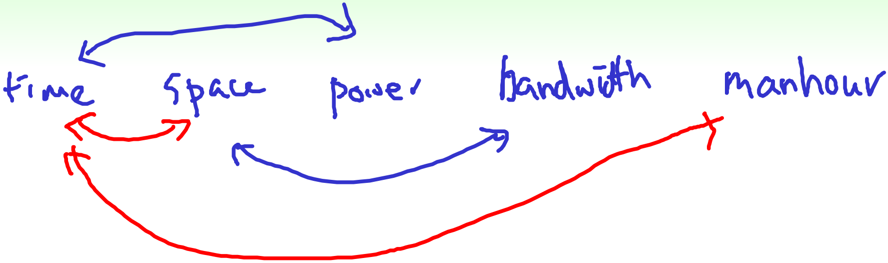
use properly

↓

good program

use storage/computation resources properly ⟹ good program

Proper Use: Tradeoff of Different Factors



understand tradeoff ⟹ good program

Different Tradeoff on Different Platforms

parallel transmission/computation

important to learn other CS subjects

Programming $\neq$ Coding

requirement
analysis
design
refinement & coding
verification: proof/test/debug

programming :: building house $\sim$ coding :: construction work

C Programming versus DSA

	C	& DSA
req.	simple	simple
analysis	simple	simple
design	simple	more
coding	***	**
proof	none	some
test	simple	**
debug	***	**

moving from coding to designing