How to solve it

Prof. Y C Cheng
2012/09/19

國立台北科技大學 資訊工程系
National Taipei University of Technology
Dept of Comp Sci and Info Engr
How to solve it

• George Polya wrote a book on the method for solving mathematical problems.
• The book has been very successful. In fact, so successful that people from other areas borrowed the notions behind “How to solve it,” including
  – architecture,
  – software development,
  – artificial intelligence, etc.
Four steps in *How to solve it*

- Understanding the problem
  - What is the unknown? What are the data? What is the condition?
- Devising a plan
  - E.g., Have you solved a similar problem before?
- Carrying out the plan
  - Requires care and patience given that you have the necessary skills
- Looking back
  - Generalize, improve, etc.
Planting trees along a pond

• A pond with a circumference of 100m long. If we were to plant a tree for every two meters around the pond, how many trees are planted?
Understand the problem

How many trees?
Devise a plan: analogy

Ans: 5
Carry out the plan

\[ 10/2 = 5 \quad \Rightarrow \quad 100/2 = 50 \]
Looking back: can I do better?

• 公式讓你算快一點？
  - 直線
    • 兩端種
    • 一端種
    • 一端種、一端不種
  - 圓形、矩形

• 真的較好？
How to solve it in programming

• Understanding the problem
  – What is the requirement? What are the data? What are the constraints?

• Devising a plan
  – Write the main program
  – List the tasks required to make the main program work

• Carrying out the plan
  – Code it up, Test it (require knowledge of C++ features, which we learn just-in-time.)

• Looking back
  – Can I improve the program? Make code cleaner and easier to read; make it run faster; make it friendlier to use;
Understand the problem

• Compute the dot product of two vectors of the same dimension. For example,

\[(1, 0) \times (1, 1) = 1\]
\[(1, 1, 0) \times (0, 1, 1) = 1\]
\[(1,0)\times(1,1,0)\text{ is illegal}\]

• What is a vector? Ordered \(n\)-tuple of real numbers

• What is the dimension of a vector? \(\text{Dim} (1, 1, 0) = 3\)

• What is dot product?

\[\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n;\]

• Can we compute dot product for vectors of different dimensions?
Devise a plan – list the tasks

1. Write the main program
2. Define a n-dimensional vector with given values for its components
3. Print a vector.
4. Compute the dot product of two vectors of the same dimension
5. Print a dot product computation.
Carry out the plan

• Write code for each of the tasks. Which first?
  – Task 1: Write the main program
  – Make it executable
  – As you complete additional tasks, compile, execute, and fix main if necessary until main executes
    (looking back)
  – And so on until you program does what it asked.

• Next slide reflects main after completing Task 2 and Task 3:
Task1: write the main program (at completion on T2 & T3)

```c
int main(int argc, char *argv[]) {
    double v1[2] = {1.0, 0.0};
    double v2[2] = {1.0, 1.0};
    double v3[3] = {1,1,0};
    double v4[3] = {0,1,1};
    printVector(v1,2);
    printVector(v2,2);
    printVector(v3,3);
    printVector(v4,3);
    // oops
    printVector(v1,3);
    printVector(v3,2);
```
Task 3: Print a vector

- Task 3 is created partly to check Task 2

```cpp
void printVector(double v[], int dim)
{
    cout << "(";
    for (int i=0; i< dim-1; ++i)
        cout << v[i] << ",";
    cout << v[dim-1] << ")" << endl;
}
```
Task 3 - testing

- Test every task
- Be picky (but with reason); ask for trouble!
Look back – can we do better?

- Double arrays as mathematical vectors? Hard to swallow??

```java
16  double v1[2] = {1.0, 0.0};
17  double v2[2] = {1.0, 1.0};
18
19  double v3[3] = {1,1,0};
20  double v4[3] = {0,1,1};
```

- Can we write vector so that it carries its own dimension? e.g.

  ```java
  printVector(v1);
  ```
int numDays = 365;
Int & j = numDays;

int * p = &numDays;

Reference (j) is just an alias (of numDays)
Model of array

int a[4] = {2,3,5,7};

What is the type of a?

1. int
2. int *
3. const int *
4. int * const

A constant pointer to integers

const double pi = 3.14159;
const double *const pi_ptr = &pi;
Create new tasks if you are not satisfied.

• Add the following tasks:
  – Define vectors that carries its own dimensionality.
• To accomplish the task you need more from C++
  – struct
  – Class
    • Data members
    • Member functions: constructor
    • Dynamic memory allocation
Class Vector

7(2). Define n-dim vectors given components that carries its own dimensionality

7.1 data members and constructor
7.2 use of heap (dynamic memory allocation)
7.3 destructor
7.4 proper encapsulation - private data members and public member functions
Staging and rework

• Staging (incremental): you cannot complete all tasks at once, you must work on them one by one
  – Do your work so that each task is executable and verifiable: we did tasks 1, 2, …, 7, one at a time

• Rework (iterative): sometimes you do not have the best solution in one try
  – Look to get a solution, then improve it. (what we have done in Task 7, which is a rework of task 2 to convert to object version.)
Concluding remarks

• Always do the four steps
• In particular, write down the tasks of steps 2 in a file, say in “todo.txt.”
• Go through the tasks in “todo.txt” one by one
  – Code it
  – Compile it (comment out unfinished lines in main to make it work)
  – Test it (using main or CppUnitLight test case)
• Until all tasks are done and results are correct
• Look for improvement opportunities