Design of Algorithms

Hwansoo Han
Problems to Programs

- Problems NOT for computers
  - "Gourmet" recipe
  - World peace

- Problems for computers
  - Formalized in (mathematical) models
  - Linear equations, differential equations
  - Symbol and text processing
Algorithms

- Ways to find solutions for problems
  - Problems in mathematical models

- Algorithm is a finite sequence of instructions

- Instructions have clear meaning
  - Can be performed within a finite amount of effort in a finite length of time
Mathematical Model (Example)

- Traffic light in an intersection of roads
Mathematical Model (Example)

- 13 Possible turns
Mathematical Model (Example)

- Incompatible turns
  - Modeled in a graph
Mathematical Model (Example)

- Incompatible turns
  - Modeled in a table
Graph Coloring as a Solution

- Graph coloring
  - Assign each color to a node (vertex), but no two nodes connected by an edge can have the same color

- Use as few colors as possible
  - Traffic light problem can be solved by graph coloring
  - But, graph coloring is \textit{NP-complete} problem

- Small graph as ours can be solved by exhaustive tries
  - For a large graph, we need an efficient solution
Heuristic for Graph Coloring

- **Greedy algorithm**
  - Color as many nodes as possible with the first color, then color as many uncolored nodes as possible with the second color, and so on.

- **Steps to color with a new color**
  1. Select some uncolored node and color it with the new color.
  2. For each uncolored nodes, check if it has an edge connected to any node already colored with the new color. If not, color the node with the new color.
Graph Coloring Example

- 2 colors
Traffic Light Example

- Graph coloring
Algorithm in Pseudo-Language

- Algorithm is presented in pseudo-language
  - Not sufficiently clear in the initial version

- Refinement to a real language
Greedy Algorithm in Pseudo-Language

- Pascal like pseudo-language

```pascal
procedure greedy ( var G: GRAPH; var newclr: SET );
{ greedy assigns to newclr a set of vertices of G that may be given the same color }
begin
(1) newclr := Ø;
(2) for each uncolored vertex v of G do
(3)   if v is not adjacent to any vertex in newclr then begin
(4)      mark v colored;
(5)      add v to newclr
   end
end; { greedy }
```
Refinement (1)

(3) if ... is refined

procedure greedy ( var G: GRAPH; var newclr: SET );
begin
    newclr := Ø;
    for each uncolored vertex v of G do begin
        found := false;
        for each vertex w in newclr do
            if there is an edge between v and w in G then
                found := true;
        if found = false then begin
            { v is adjacent to no vertex in newclr }
            mark v colored;
            add v to newclr
        end
    end
end; { greedy }
Refinement (2)

- for each ... is refined

procedure greedy ( var G: GRAPH; var newclr: LIST );
{ greedy assigns to newclr those vertices that may be
given the same color }

var
found: boolean;
v, w: integer;
begin
newclr := ∅;
v := first uncolored vertex in G;
while v <> null do begin
found := false;
w := first vertex in newclr;
while w <> null do begin
if there is an edge between v and w in G then
found := true;
w := next vertex in newclr
end;
if found = false do begin
mark v colored;
add v to newclr
end;
v := next uncolored vertex in G
end
end; { greedy }
Abstract Data Type (ADT)

- Abstract data type
  - Mathematical model + operations

ADT vs. Procedure

- Procedure
  - Generalization of operators (+, -, ...)

- ADT
  - Generalization of primitive types (int, float, ...)
Abstract Data Type (ADT)

- Example of ADT
  - LIST (of integer)
    - MAKENULL() : make a list empty
    - FIRST(), NEXT()
    - INSERT()

- Class in C++
  - Pseudo-language
    
    \[w := \text{next vertex in } newclr\]

  - C++
    
    ```cpp
    class List {
    ...
    int next();
    ...
    }
    ```
    
    List newclr;
    
    w = newclr.next();