11. Greedy Approach

Goals

- Learn greedy approach.
- Understand the cases where greedy algorithm leads to optimal solution and the cases where greedy algorithm doesn't lead to optimal solution.
- Learn some greedy algorithms.

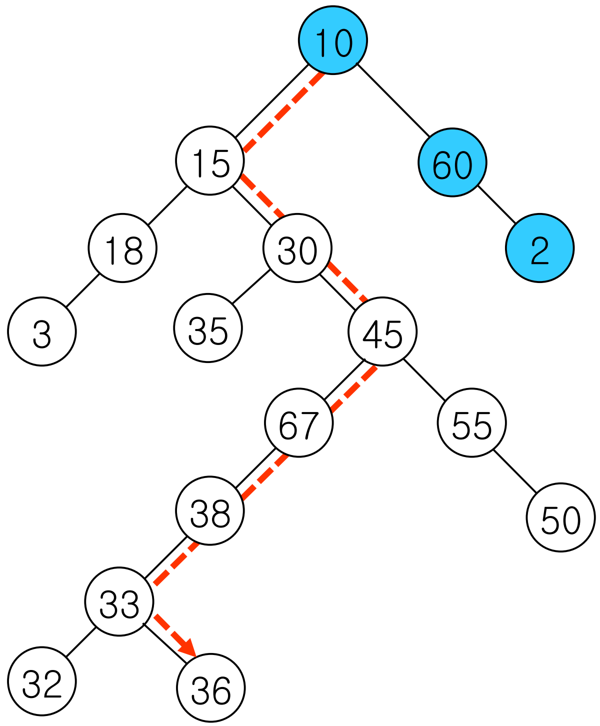
Greedy Approach

- It makes a locally optimal choice repeatedly.
- It may not arrive at a globally optimal solution.
- Sometimes locally optimal choices lead to a globally optimal solution.

```
do {
        make a locally optimal choice
} until (a solution is obtained)
```

Globally Optimal Solution Not Obtained

Find the root-to-leaf path in which the sum of weights is maximum
When we arrive at a node, we can see only weights of its children.

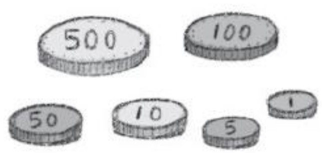


Greedy algorithm: always choose the child with larger weight

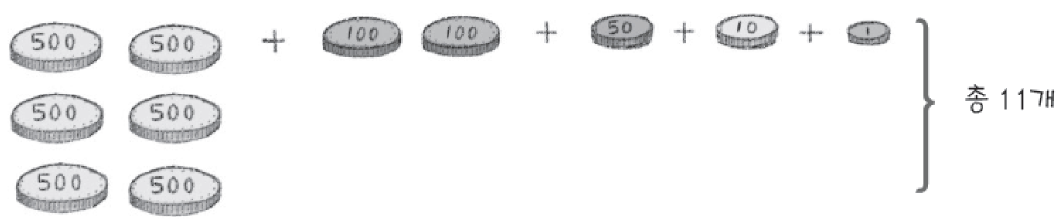
Globally Optimal Solution Obtained/Not Obtained

Coin Changing Problem

Coin denominations



Target: 3,261

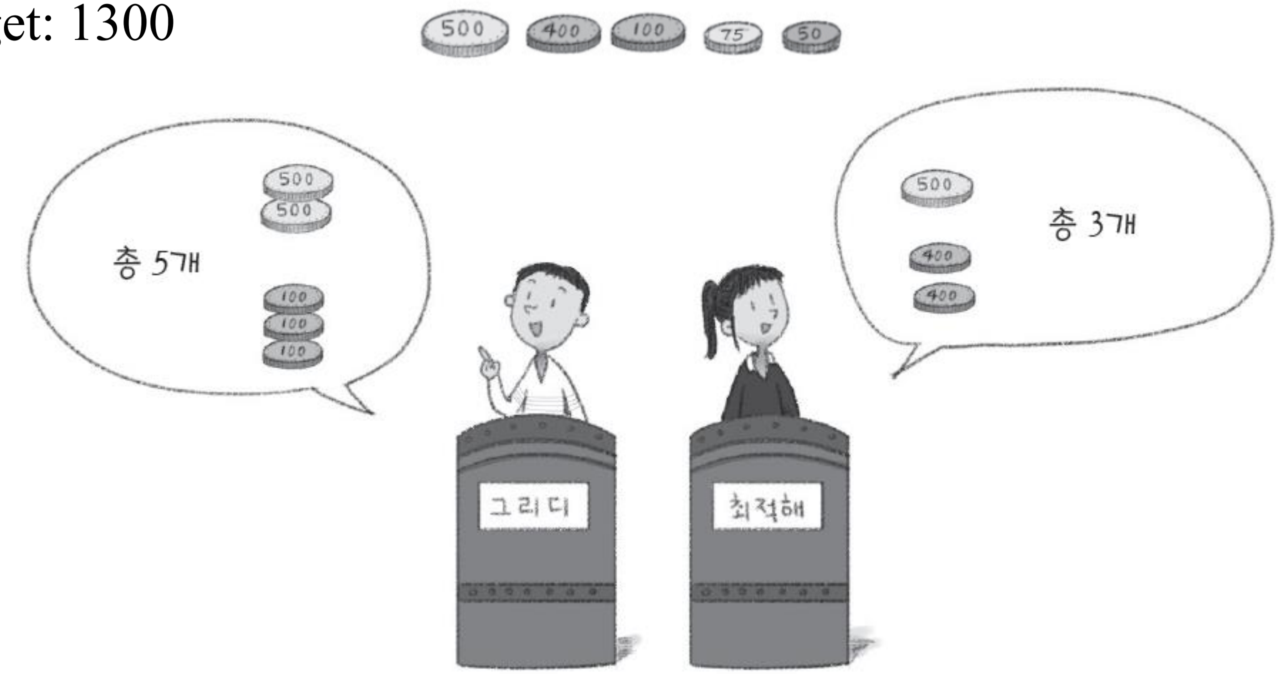


Greedy algorithm: starting from largest denomination, always choose maximum number of coins with the denomination

If each denomination is a multiple of previous denomination, greedy algorithm yields an optimal solution

If a denomination is not a multiple of previous denomination, greedy algorithm does not yield an optimal solution

Target: 1300



Structure of Greedy Algorithm

Greedy(C)
// C: set of input elements
{
    $S \leftarrow \emptyset$;
    while ($C \neq \emptyset$ **and** S is not a solution) {
        $x \leftarrow$ locally optimal element in C;
        remove x from C;   // $C \leftarrow C - \{x\}$
        if (x can be added to S) **then** $S \leftarrow S \cup \{x\}$;
    }
    if (S is a solution) **then return** S;
        else return "no solution!";
}

Greedy Algorithm

Prim's algorithm for minimum spanning tree

```
Prim (G, r)
{
   S ← Φ ;
   Mark r visited, and insert r into S;
   while (S≠V) {
       Find edge (x, y) with minimum weight between S and V-S;    ▷ x∈S, y∈V-S
       Mark y visited, and insert y into S;
   }
}
```

greedy

Activity Selection Problem

- One meeting room
- Many departments make requests for meeting room
  - n activities $\{a_1, a_2, \dots, a_n\}$
  - Each activity has start time s_i and finish time f_i with interval $[s_i, f_i)$
- Greedy choices
  - Choose activity with shortest spending time
  - Choose activity with earliest start time
  - Choose activity with earliest finish time

Only this one yields optimal solution

Activity Selection Problem

- Let a_k be activity with earliest finish time. a_k is included in an optimal solution.
- Proof: Let A be an optimal solution, and a_j be the activity with earliest finish time in A. If $a_j = a_k$, we are done. If $a_j \neq a_k$, $A - \{a_j\} \cup \{a_k\}$ becomes an optimal solution.

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16

- Solution: $\{a_3, a_9, a_{11}\}$
- Optimal solution: $\{a_1, a_4, a_8, a_{11}\}, \{a_2, a_4, a_9, a_{11}\}$

Activity Selection Problem

```
Greedy(s,f)
{
    Assume that activities are sorted by finish time
    A ← {a_1}
    j ← 1
    for i ← 2 to n
        if s_i ≥ f_j  then {
            A ← A ∪ {a_i};
            j ← i;
        }
    return A
}
```

Thank you