CSE 373/L-13/24.03.2024/

Chapter-16

Greedy Algorithms

* Greedy algorithm can be used for many optimization problems, but not always.

* Greedy algorithm always makes choice that looks best at the moment. Don't think about future.
  - it hope that, a locally optimal choice will lead to a globally optimal solution
    - for some problems, it works.

* Two ingredients that need to prove ~~to~~ for a greedy strategy:

  (i) Greedy-choice property → choose best choice at the moment
  → don't think about future sub problem.

  (ii) Optimal substructure

⇒ and we need to prove that a greedy choice at each step yields a globally optimal solution.

* Optimal Substructure: a problem exhibits optimal substructure if an optimal solution to the problem contains optimal solution to sub problem.

An activity selection problem:

Definition:

- Scheduling a resource among competing activities.

Elaboration:

$\Rightarrow S = 1, 2, 3, \ldots n$ activities

activity a_i has,

start time s_i
finish time f_i

$0 \le s_i \le f_i \le \infty$

Resource Opening time

Resource Closing time

Compatibility:

activity $a_i \Rightarrow [s_i, f_i)$
activity $a_j \Rightarrow [s_j, f_j)$

$s_i \ge f_j$
or
$s_j \ge f_i$

Goal:

- to select a maximum size set of mutually ~~manually~~ compatible activities.

$S \Rightarrow$

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

$\Rightarrow$ finish time is sorted already.

* Optimal solution can be multiple:

$\{a_1, a_4, a_8, a_{11}\}$

$\{a_2, a_4, a_8, a_{11}\}$

$\{a_2, a_4, a_9, a_{11}\}$

Greedy algorithm will give one of them.

* Property of greedy algorithm here,

- choose one of them which must finish first.
- the next selected activity is always the one with the earliest finish time.
  - it is a greedy choice in the sense that it leaves as much as opportunity as possible for the remaining activities to be scheduled.
  - Thus it ~~maximum~~ maximize the amount of ~~time~~ unscheduled time remaining.

* Algorithm:

GREEDY-ACTIVITY-SELECTOR(s, f)

$\Theta(n)$

```
n = s.length
A = {a1}
k = 1
for m = 2 to n
    if s[m] ≥ f[k]
        A = A ∪ {am}
        k = m
return A
```

total time $= \underbrace{O(n \lg n)}_{\text{for sort}} + \Theta(n)$

$= O(n \lg n)$

Operation:

$A = \{a_1, a_4, a_8, a_{11}\}$ $S = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}\}$

$A' = \{a_4, a_8, a_{11}\} \leftarrow S' = \{a_4, a_6, a_7, a_8, a_9, a_{11}\}$

$A'' = \{a_8, a_{11}\} \leftarrow S'' = \{a_8, a_9, a_{11}\}$

$= \{a_9, a_{11}\}$ $S''' = \{a_{11}\}$

$A''' = \{a_{11}\}$ $S^{iv} = \{\phi\}$

Why it is greedy?

- it leaves as much opportunity as possible for the remaining activities to be scheduled.

- The greedy choice is the one that maximizes the amount of unscheduled time remaining.

Why this algorithm is optimal?

- properties
  - satisfies the greedy choice property
  - has the optimal substructure property

⊛ Greedy choice property:

⇒ Show there is an optimal solution that start with a greedy choice. $(k=1)$

⇒ Suppose $A \subseteq S$ in an optimal solution.

- Order the activities in A by finish time. The first activity in A is k.

  ⇒ if $k=1$, the schedule A begins with greedy choice.

  ⇒ if $k \neq 1$, show that there is an optimal solution B to S that begins with the greedy choice, activity 1.

⇒

$A'' = \{a_9, a_{11}\}$; $S'' = \{a_8, a_9, a_{11}\}$

activity 1

if, $A'' = \{a_8, a_{11}\}$ also a optimal solution, then its greedy choice.

⇒ Let, $B = A - \{k\} \cup \{1\}$

- Because $f_1 \leq f_k$ → activities in B are disjoint
- since B has the same number of activities as A
- Thus, B is optimal.

Optimal Substructure:

⇒ Once the greedy choice of activity 1 is made, the problem reduces to finding an optimal solution for the activity selection problem over those activity in S that are compatible with activity 1.

- if A is optimal solution to S, the $A' = A - \{1\}$ is optimal to $S' = \{i \in S : s_i \geq f_1\}$

⇒ why?

⇒ if we find another optimal B' to S' with more activities than A' then, adding activity 1 to B' would yield a solution B to S with more activity than A, will contradict the optimality of A

$A = \{a_1, a_4, a_8, a_{11}\} \Rightarrow 4$

$A' = \{a_4, a_8, a_{11}\} \Rightarrow 3$

if, $B' = \{a_4, a_8, a_{11}, \square\} \Rightarrow 4$ is possible

then $B = \{a_1, a_4, a_8, a_{11}, \square\} \Rightarrow 5$

And $B > A$

But B' is not possible

then A is the optimal solution. (Proved) by contradict.

⇒ After each greedy choice is made, we are left with an optimization problem of the same ~~from as~~ form as the original problem.

- By induction on the number of choice made, making the greedy choice at every step produces an optimal solution.

Next class Quize-2

Ch-6,7,8

L-14 / 31.03.2024 /

Quiz-2