Greedy algorithms

A typical algorithm:

A typical algorithm:

1. Make some decision(s)

A typical algorithm:

1. Make some decision(s)
2. Problem is broken into k subproblems $(n_1, \ldots, n_k)$

A typical algorithm:

1. Make some decision(s)
2. Problem is broken into k subproblems $(n_1, \ldots, n_k)$
3. Solve k subproblems recursively

Greedy algorithms

A typical algorithm:

1. Make some decision(s)
2. Problem is broken into k subproblems $(n_1, \ldots, n_k)$
3. Solve k subproblems recursively
4. Combine decision(s) from (1) with solutions from (3), to output solution

GREEDY:

Greedy algorithms

GREEDY:

1. Make greedy choice g

GREEDY:

1. Make greedy choice g
2. Problem is reduced into one subproblem

Greedy algorithms

GREEDY:

1. Make greedy choice g
2. Problem is reduced into one subproblem
3. Solve subproblem recursively $\Rightarrow SOL_{sub}$

GREEDY:

1. Make greedy choice g
2. Problem is reduced into one subproblem
3. Solve subproblem recursively $\Rightarrow SOL_{sub}$
4. Combine choice from (1) with solution from (3), to output solution SOL

Greedy algorithms

GREEDY:

1. Make greedy choice g
2. Problem is reduced into one subproblem
3. Solve subproblem recursively $\Rightarrow SOL_{sub}$
4. Combine choice from (1) with solution from (3), to output solution SOL

Correctness:

Theorem

If we have that

1. *greedy choice is part of an OPT solution*
2. *$SOL = g \cup SOL_{sub}$ is a feasible solution*

then SOL is an optimal solution (i.e., greedy alg is correct).

Greedy algorithms: Correctness

Theorem

If we have that

1. *greedy choice is part of an OPT solution*
2. $SOL = g \cup SOL_{sub}$ *is a feasible solution*

then SOL *is an optimal solution (i.e., greedy alg is correct).*

Greedy algorithms: Correctness

Theorem

If we have that

1. *greedy choice is part of an OPT solution*
2. $SOL = g \cup SOL_{sub}$ *is a feasible solution*

then SOL is an optimal solution (i.e., greedy alg is correct).

Proof: By induction on the input size:

Greedy algorithms: Correctness

Theorem

If we have that

1. *greedy choice is part of an OPT solution*
2. $SOL = g \cup SOL_{sub}$ *is a feasible solution*

then SOL is an optimal solution (i.e., greedy alg is correct).

Proof: By induction on the input size:

- $n = 1$: From (1), $SOL = g = OPT$.

$\square$

Greedy algorithms: Correctness

Theorem

If we have that

1. *greedy choice is part of an OPT solution*
2. $SOL = g \cup SOL_{sub}$ *is a feasible solution*

then SOL is an optimal solution (i.e., greedy alg is correct).

Proof: By induction on the input size:

- $n = k$: Up to size k, $\textsc{Greedy}$ computes OPT.

$\square$

Greedy algorithms: Correctness

Theorem

If we have that

1. *greedy choice is part of an OPT solution*
2. $SOL = g \cup SOL_{sub}$ *is a feasible solution*

then SOL is an optimal solution (i.e., greedy alg is correct).

Proof: By induction on the input size:

- $n = k + 1$: Because of inductive step, SOL_{sub} is optimal solution of the subproblem in $\textsc{Greedy}$.

Greedy algorithms: Correctness

Theorem

If we have that

1. *greedy choice is part of an* OPT *solution*
2. $SOL = g \cup SOL_{sub}$ *is a feasible solution*

then SOL *is an optimal solution (i.e., greedy alg is correct).*

Proof: By induction on the input size:

- $n = k + 1$: Because of inductive step, SOL_{sub} is optimal solution of the subproblem in GREEDY.
  If we "contract" g in OPT in (1), and remain feasible,
  $$OPT = g \cup OPT', \qquad (1)$$
  we get an optimal solution OPT' of the subproblem in GREEDY, i.e.,
  $$OPT' = SOL_{sub} \qquad (2)$$

Theorem

If we have that

1. *greedy choice is part of an OPT solution*
2. *$SOL = g \cup SOL_{sub}$ is a feasible solution*

then SOL is an optimal solution (i.e., greedy alg is correct).

Proof: By induction on the input size:

- $n = k + 1$: Because of inductive step, SOL_{sub} is optimal solution of the subproblem in GREEDY.
  If we "contract" g in OPT in (1), and remain feasible,

$$OPT = g \cup OPT', \tag{1}$$

  we get an optimal solution OPT' of the subproblem in GREEDY, i.e.,

$$OPT' = SOL_{sub} \tag{2}$$

  Cobining (1), (2) we get the theorem.

Greedy algorithms: Correctness

Theorem

If we have that

1. *greedy choice is part of an OPT solution*
2. $SOL = g \cup SOL_{sub}$ *is a feasible solution*

then SOL is an optimal solution (i.e., greedy alg is correct).

Observations:

Greedy algorithms: Correctness

Theorem

If we have that

1. *greedy choice is part of an OPT solution*
2. $SOL = g \cup SOL_{sub}$ *is a feasible solution*

then SOL is an optimal solution (i.e., greedy alg is correct).

Observations:

- Usually (2) is trivial, GREEDY is designed to satisfy it.

Greedy algorithms: Correctness

Theorem

If we have that

1. *greedy choice is part of an OPT solution*
2. $SOL = g \cup SOL_{sub}$ *is a feasible solution*

then SOL is an optimal solution (i.e., greedy alg is correct).

Observations:

- Usually (2) is trivial, GREEDY is designed to satisfy it.
- Here are two approaches to prove (1):

Greedy algorithms: Correctness

Theorem

If we have that

1. *greedy choice is part of an OPT solution*
2. $SOL = g \cup SOL_{sub}$ *is a feasible solution*

then SOL is an optimal solution (i.e., greedy alg is correct).

Observations:

- Usually (2) is trivial, GREEDY is designed to satisfy it.
- Here are two approaches to prove (1):
  1. Show that GREEDY is always ahead (i.e., partial solution built with greedy choices is better than *any* other partial solution, up to the end).

Theorem

If we have that

1. *greedy choice is part of an OPT solution*
2. $SOL = g \cup SOL_{sub}$ *is a feasible solution*

then SOL is an optimal solution (i.e., greedy alg is correct).

Observations:

- Usually (2) is trivial, GREEDY is designed to satisfy it.
- Here are two approaches to prove (1):
  1. Show that GREEDY is always ahead (i.e., partial solution built with greedy choices is better than *any* other partial solution, up to the end).
  2. Show that from *any* OPT solution (where greedy choice g may not be the first one), we can derive another optimal solution OPT' where g *is* its first choice, performing a series of exchanges.