

19.2

Greedy Algorithms: Tools and Techniques

What is a Greedy Algorithm?

No real consensus on a universal definition.

Greedy algorithms:

- 1 make decision incrementally in small steps without backtracking
- 2 decision at each step is based on improving local or current state in a myopic fashion without paying attention to the global situation
- 3 decisions often based on some fixed and simple priority rules

What is a Greedy Algorithm?

No real consensus on a universal definition.

Greedy algorithms:

- 1 make decision incrementally in small steps without backtracking
- 2 decision at each step is based on improving local or current state in a myopic fashion without paying attention to the global situation
- 3 decisions often based on some fixed and simple priority rules

What is a Greedy Algorithm?

No real consensus on a universal definition.

Greedy algorithms:

- 1 make decision incrementally in small steps without backtracking
- 2 decision at each step is based on improving local or current state in a myopic fashion without paying attention to the global situation
- 3 decisions often based on some fixed and simple priority rules

Pros and Cons of Greedy Algorithms

Pros:

- ① Usually (too) easy to design greedy algorithms
- ② Easy to implement and often run fast since they are simple
- ③ Several important cases where they are effective/optimal
- ④ Lead to a first-cut heuristic when problem not well understood

Cons:

- ① **Very often** greedy algorithms don't work. Easy to lull oneself into believing they work
- ② Many greedy algorithms possible for a problem and no structured way to find effective ones

CS 374: Every greedy algorithm needs a proof of correctness

Pros and Cons of Greedy Algorithms

Pros:

- ① Usually (too) easy to design greedy algorithms
- ② Easy to implement and often run fast since they are simple
- ③ Several important cases where they are effective/optimal
- ④ Lead to a first-cut heuristic when problem not well understood

Cons:

- ① **Very often** greedy algorithms don't work. Easy to lull oneself into believing they work
- ② Many greedy algorithms possible for a problem and no structured way to find effective ones

CS 374: Every greedy algorithm needs a proof of correctness

Pros and Cons of Greedy Algorithms

Pros:

- ① Usually (too) easy to design greedy algorithms
- ② Easy to implement and often run fast since they are simple
- ③ Several important cases where they are effective/optimal
- ④ Lead to a first-cut heuristic when problem not well understood

Cons:

- ① **Very often** greedy algorithms don't work. Easy to lull oneself into believing they work
- ② Many greedy algorithms possible for a problem and no structured way to find effective ones

CS 374: Every greedy algorithm needs a proof of correctness

Greedy Algorithm Types

Crude classification:

- 1 **Non-adaptive:** fix some ordering of decisions a priori and stick with the order
- 2 **Adaptive:** make decisions adaptively but greedily/locally at each step

Plan:

- 1 See several examples
- 2 Pick up some proof techniques

Greedy Algorithm Types

Crude classification:

- ① **Non-adaptive:** fix some ordering of decisions a priori and stick with the order
- ② **Adaptive:** make decisions adaptively but greedily/locally at each step

Plan:

- ① See several examples
- ② Pick up some proof techniques

THE END

...

(for now)