

*The exam will be in class (one hour) and closed book. You may use one page of notes (one side of an 8.5×11" piece of paper, hard copy required), which will be handed in with the exam.*

*It is strongly recommended that you treat this as a practice exam and work on it in a similar environment.*

---

1. There are  $n$  packets of cards, each of which contains a subset of the available cards. Assume that you can see through the packaging, so you know the contents of each packet. You want to buy as few packets as possible in order to get the full set of cards.
  - (a) Give a plausible greedy strategy for this problem: The next packet to buy is... [fill in the rest of this sentence]
  - (b) Give a counterexample to illustrate why this greedy strategy doesn't work.
2. A set of highly collectable toys has been released. Due to high demand, customers are only allowed to purchase one toy per day and only one of each kind of toy in total. Prices are also rising — each toy costs  $p_i^d$  where  $d \geq 0$  is the day the toy is purchased and  $p_i$  is the base price for toy  $i$ . Your goal is to purchase all of the toys as cheaply as possible.

Which of the following, if any, would be appropriate loop invariants for showing that purchasing the toys in order of decreasing base price (that is, highest  $p_i$  first) is the right strategy? For each, either write “good” indicating that it is an appropriate loop invariant or explain why it is not.

- (a) The first  $k$  toys purchased are those with the highest base price.
- (b) The next toy purchased is the one with the highest base price amongst those left to buy.
- (c) After  $k$  toys have been purchased, the total amount paid so far is the least possible for those toys.
- (d) After  $k$  toys have been purchased, the total amount paid so far is the lowest possible for any set of  $k$  toys.
- (e) After  $k$  toys have been purchased, the total amount paid so far is no higher than was paid for those toys in the optimal solution.
- (f) After  $k$  toys have been purchased, the total amount paid so far is no higher than was paid for the first  $k$  toys purchased in the optimal solution.
- (g) After  $k$  toys have been purchased, the number of days needed to purchase those toys is no more than is needed in the optimal solution.

3. A *clique*  $C$  is a set of vertices of a graph  $G$  such that every pair of vertices in  $C$  is connected in  $G$ . The CLIQUE problem asks whether, given an undirected graph  $G$  and an integer  $k$ ,  $G$  contains a clique of size at least  $k$ .

A *vertex cover*  $V'$  is a set of vertices of a graph  $G$  such that every edge of  $G$  has at least one endpoint in  $V'$ . The VERTEX COVER problem asks whether, given an undirected graph  $G$  and an integer  $k$ , there is a vertex cover of size at most  $k$ .

- (a) Explain why VERTEX COVER is in NP.
- (b) CLIQUE is known to be NP-complete. To show that VERTEX COVER is also NP-complete, should you reduce it to CLIQUE or CLIQUE to it? Explain. (You do not have to give the reduction, just which way the reduction should go and why.)
- (c) The BOX DEPTH problem asks whether, given a set of  $n$  axis-aligned rectangles in the plane, there is a subset of at least  $k$  rectangles containing a common point.

There is a polynomial-time reduction from BOX DEPTH to CLIQUE. There is also a polynomial-time algorithm for solving BOX DEPTH directly. Explain why the existence of an efficient algorithm for BOX DEPTH does not imply that  $P = NP$  even though CLIQUE is known to be NP-complete.

4. You are standing at a scenic overlook in the Adirondacks, gazing out at the view — endless rows of mountains. Wanting to know what you are looking at, you consult a peak finder app on your phone which labels those peaks that stick up high enough to be silhouetted against the sky.



Given a simplified world of triangular mountains (as shown above), determine the peaks that should be labeled by the peak finder. Each mountain  $i$  is described by an interval  $(l_i, r_i)$  indicating the left and right ends of the horizontal span of its base and a height  $h_i$  indicating the height of its peak, which can be assumed to be centered in the interval.

Develop an efficient **divide-and-conquer** algorithm for solving the problem using the divide-and-conquer development process from class — address the steps below.

- (a) Specifications.
- (b) Targets.

- (c) Patterns.
- (d) Size.
- (e) Generalize / define subproblems.
- (f) Base case(s).
- (g) Main case.
- (h) Top level: initial subproblem, setup, wrapup.
- (i) Termination: making progress, the end is reached.
- (j) Correctness: establish the base case(s), show the main case, final answer.
- (k) Time and space.

5. A golf course has a gopher problem — they just keep popping up everywhere! However, owls prey on gophers, and one fake owl can deter any gopher that pops up within a distance  $d$  left and right of its position along the fairway. Given the position of each gopher hole (as a distance along the fairway), determine where to place the fake owls so that every gopher hole is covered and the minimum number of owls is used.

Develop a **greedy** algorithm for solving the problem using the greedy development process from class — address the steps below.

- (a) Specifications.
- (b) Patterns.
- (c) Greedy choices and counterexamples.
- (d) Main steps.
- (e) Exit condition.
- (f) Setup.
- (g) Wrapup.
- (h) Termination: measure of progress, making progress, the end is reached.
- (i) Loop invariant.
- (j) Establish the loop invariant.
- (k) Maintain the loop invariant.
- (l) Final answer.
- (m) Time and space.

6. You are preparing for a multi-day backpacking trip with a group of friends. You've assembled the pile of gear but now all those items — tents, cooking gear, food, clothing, etc. — need to be packed into backpacks. Each of the  $n$  items

has a weight, and each of the  $k$  hikers in the group has a maximum amount of weight they can carry. (Different people may have different limits.) Is it possible to distribute all of the items amongst the hikers so that no one is overloaded?

Develop a **backtracking** algorithm for solving the problem using the backtracking development process from class — address the steps below.

- (a) Specifications.
- (b) Patterns.
- (c) The series of choices.
- (d) Size.
- (e) Generalize / define subproblems: partial solution, alternatives, subproblem.
- (f) Base case(s).
- (g) Main case.
- (h) Top level: initial subproblem, setup, wrapup.
- (i) What can you prune beyond just illegal alternatives for the next choice? How effective do you think this will be?
- (j) Identify whether the bound function for a branch-and-bound algorithm should be an upper bound or a lower bound. Give one possible safe bound function, and explain why it is safe. How effective do you think it will be?

7. You walk into Staples, with only  $m$  minutes to make  $n$  copies of a key document before an important meeting. The store has  $k$  copiers with the necessary features to handle your job, but they have different copying speeds and availabilities — some machines are faster than others, and there are other customers doing their own copying.

Given a projection  $n_{ij}$  of how many copies you could make on each machine  $i$  in minute  $j$ , determine whether it is possible to complete your job in time. Note that you don't have to use the same copier for the entire job, but switching from one machine to another takes a minute during which no copying is done. (You have to gather up your copies, walk over to the other machine, configure its settings, etc.)

Develop an efficient **dynamic programming** algorithm for solving the problem using the dynamic programming development process from class — address the steps below.

- (a) Specifications.
- (b) Patterns.
- (c) The series of choices.
- (d) Size.

- (e) Generalize / define subproblems: partial solution, alternatives, subproblem.
- (f) Memoization.
- (g) Base case(s).
- (h) Main case.
- (i) Top level: initial subproblem, setup, wrapup.
- (j) Algorithm.
- (k) Time and space.