resources: practice exam

Name _____

CSCI 332, Fall 2025

Exam 2

Note that this exam has three sections. They are:

1. Greedy Algorithms (30 points)
2. Divide and Conquer (35 points)
3. Dynamic Programming (35 points)

You may use a double sided 3x5 handwritten notecard of notes during the test but no other resources. If you need more space than what is given, develop your solution on scratch paper before copying your final answer to the exam paper.

Good luck!

Greedy

Given a computational problem,

- propose some greedy strategies to solve it (apply strategy)

Given a greedy algorithm,

- be able to trace through the algorithm's execution on a given input (trace)
- if incorrect, give a example input for which the alg gives wrong answer (counterexample)

- be able to analyze runtime (runtime)
- be able to understand a proof of correctness (if correct) (proof of correctness)
  - inductive exchange
  - fill in or reorder lines partial
  - identify what proof is talking about on an example

Examples:

- earliest finish time first
- Dijkstra

Divide and Conquer

Given a computational problem:

- propose some D+C strategies to solve

Given a proposed D+C algorithm:

- trace: recursion trees
- counterexample
- proof of correctness (induction)
- runtime: write recurrence for a D+C alg and solve recurrences of the form $T(n) = AT(\frac{n}{B}) + f(n)$

Via recursion trees

Examples:

- mergesort
- split Multiply
- Karatsuba's

Dynamic Programming

Given a computational problem:

- propose some DP strategies to solve it

(1) English definition of subproblem
(2) Recursive definition of subproblem
(3) How to memoize (array)
(4) DP alg (iterative)

Given a DP alg:

not memoized ↓ memoized ↓

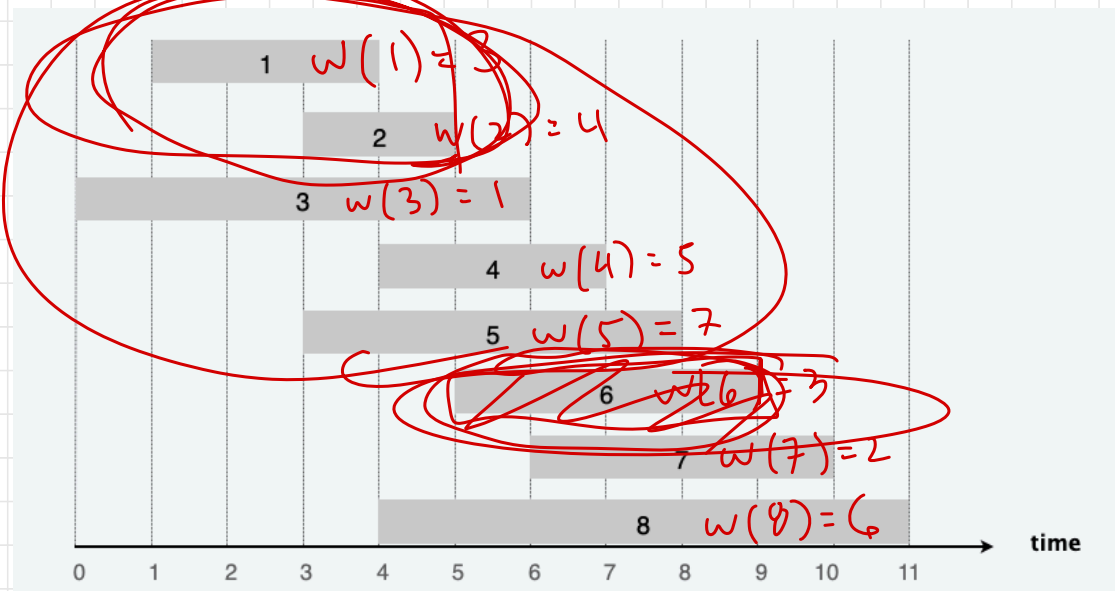
- trace (both iterative and recursive)
- counterexample
- proof of correctness (induction)
- runtime (both iterative and recursive)

Examples:

- fibo, MinCostHotel, max array sum,

max candy

weighted interval scheduling



largest set of compatible intervals

can select 2.

greedy: {1, 4}

weight of heaviest set of compatible intervals

w(1) = 3
w(4) = 5 total weight 8

{1, 8} w(1) = 3
w(8) = 6 total 9

EFTF greedy?

X

DP

① English def. of subproblems

let MaxWeight(i) be the maximum weight of any compatible set of intervals up to interval i.

② write recursive def. of subproblem

$$\text{MaxWeight}(i) = \begin{cases} w(1) & \text{if } i = 1 \\ \end{cases}$$

Notice that for interval i, I either:

- don't include it (MaxWeight(i-1))
- include i

  w(i) + MaxWeight(~~largest i~~ latest finishing j not overlapping w/ i)