CS180 Discussion

Week 1

Office Hours

Monday: 8:00 - 10:00

Outside 391 ENG VI

Lecture Recap

- Celebrity problem
- Stable Matching
- Big O Small Ω
- Amortized analysis
- Von Neumann architecture
- ~~Interval scheduling~~
- ~~Majority calculation algorithm~~

Celebrity Problem

A party of N people, only one person is known to everyone. Such a person may be present in the party, if yes, (s)he doesn’t know anyone in the party. We can only ask questions like “does A know B? “. Find the stranger (celebrity) in minimum number of questions.

Stable Matching

set of two men, {m, m′}, and a set of two women, {w, w′}. The preference lists are as follows:

m prefers w to w′.

m′ prefers w′ to w.

w prefers m′ to m.

w′ prefers m to m′.

(i) M prefers his wife to another woman W; or

(ii) W prefers her current husband over another man M.

Gale-Shapley algorithm

Initially all $m \in M$ and $w \in W$ are free
While there is a man m who is free and hasn't proposed to every woman

a. Choose such a man m
b. Let w be the highest-ranked woman in m's preference list
c. to whom m has not yet proposed;
d. If w is free then
    i. (m, w) become engaged
e. Else
    w is currently engaged to m'
    i. If w prefers m' to m then
        1. m remains free
    ii. Else
        1. w prefers m to m' (m, w) become engaged m' becomes free
f. Endif
    i. Endif

Endwhile
Return the set S of engaged pairs

Gale-Shapley algorithm

Initially all $m \in M$ and $w \in W$ are free
While there is a man m who is free and hasn't proposed to every woman

a. Choose such a man m
b. Let w be the highest-ranked woman in m's preference list
c. to whom m has not yet proposed;
d. If w is free then
    i. (m, w) become engaged
e. Else
    i. w is currently engaged to m'
    ii. If w prefers m' to m then
        1. m remains free
    iii. Else
        1. w prefers m to m'(m, w) become engaged m' becomes free
f. Endif
    i. Endif

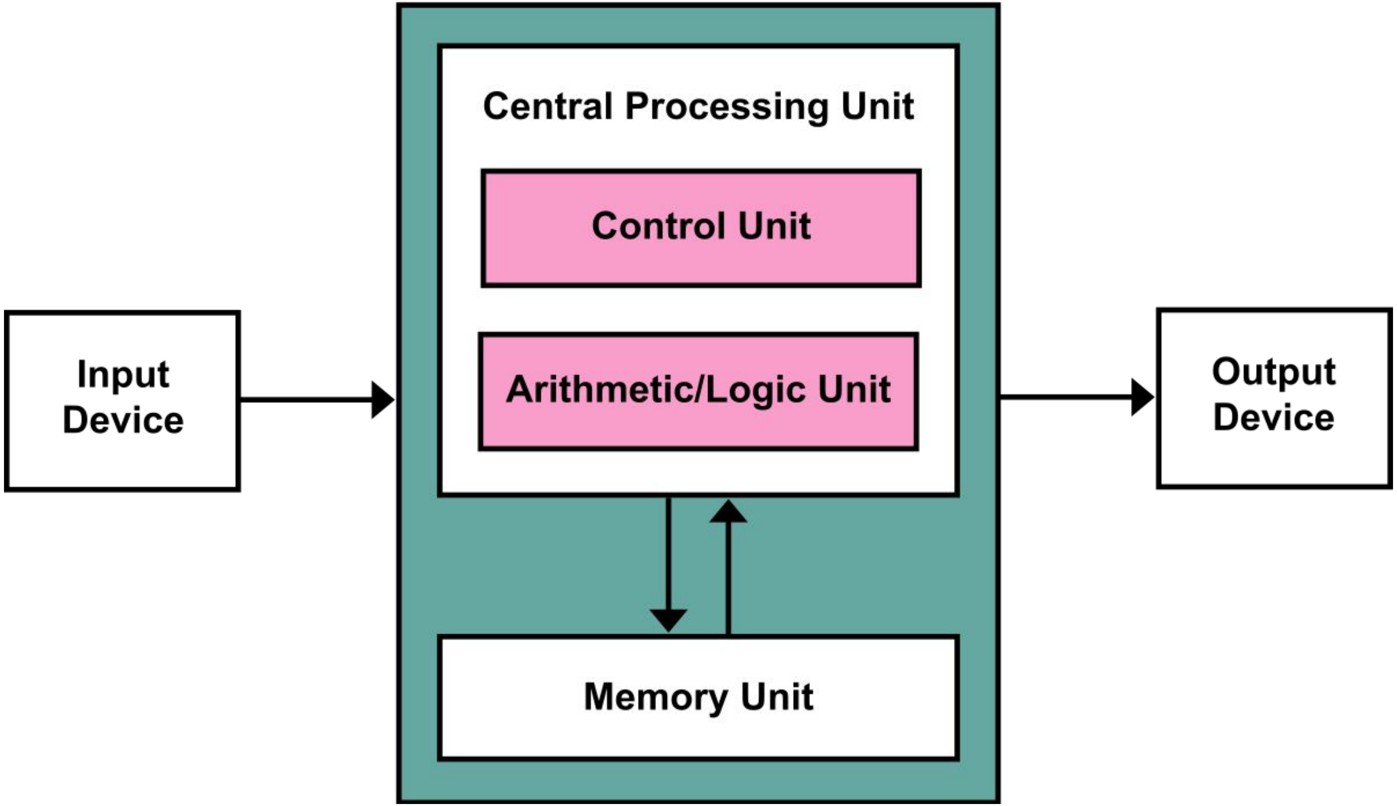
Endwhile
Return the set S of engaged pairs

$$O(n^2)$$

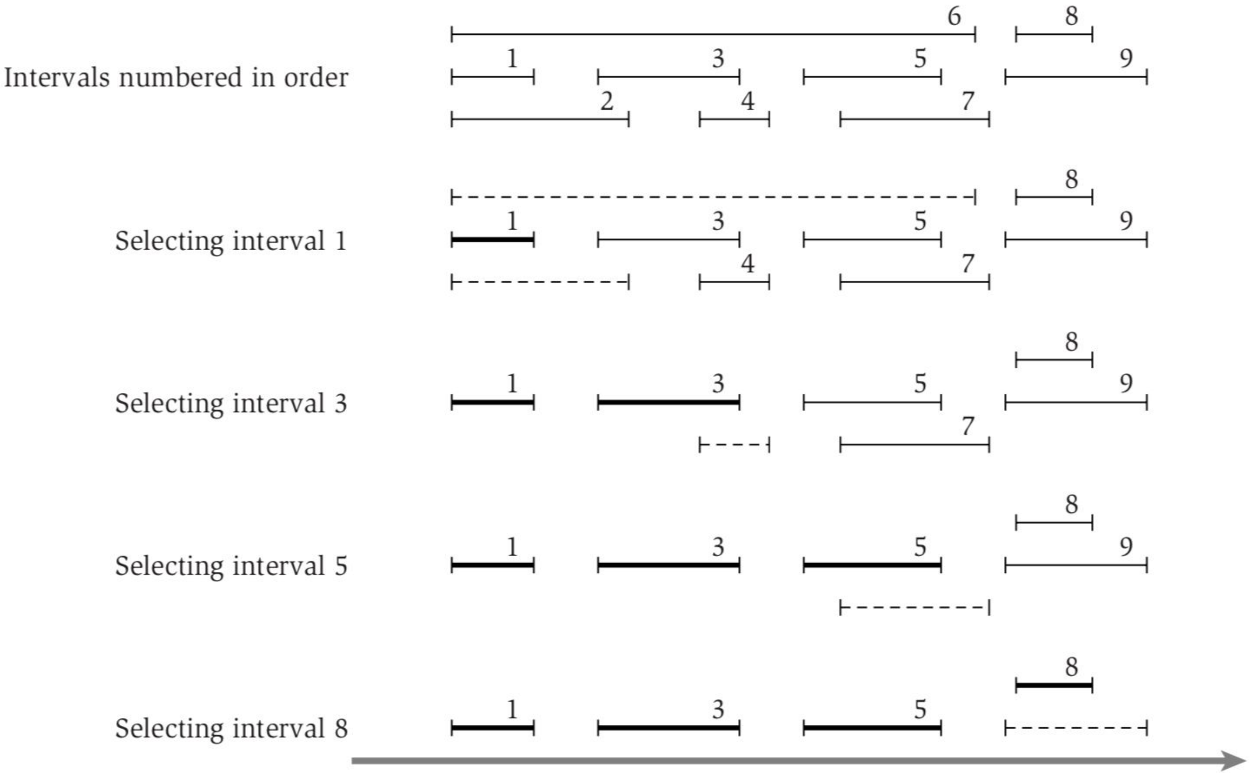
PAIRS:

(m, w), (m', w')

Von Neumann architecture

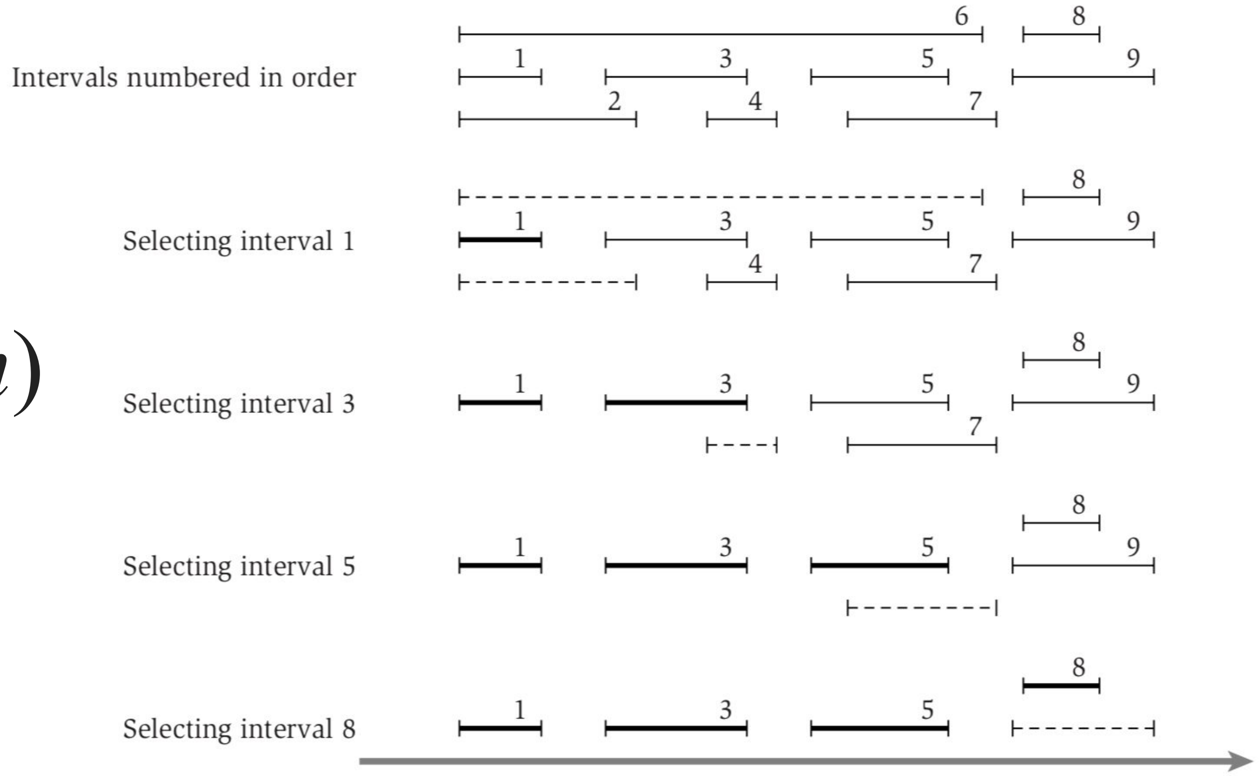


Interval scheduling

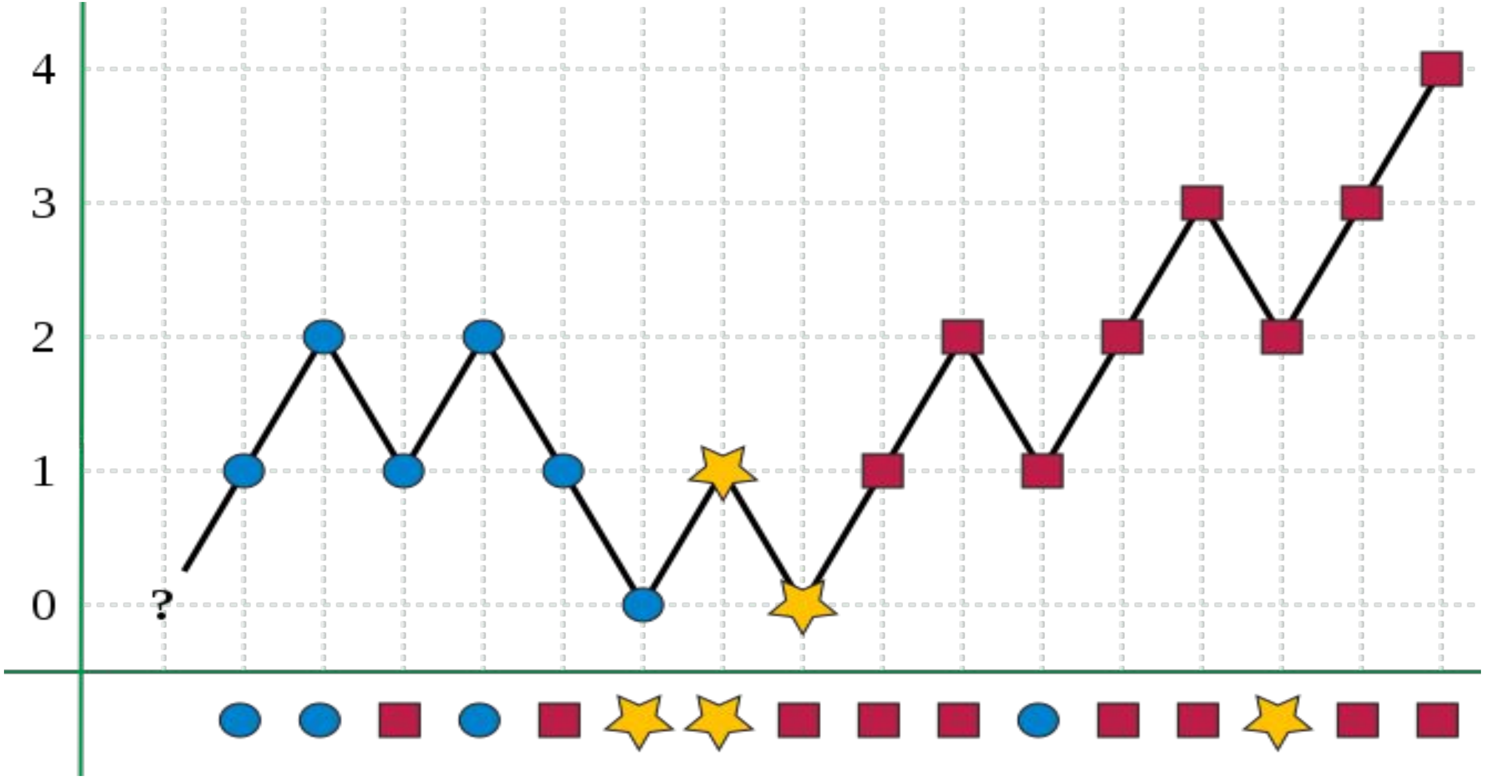


Interval scheduling

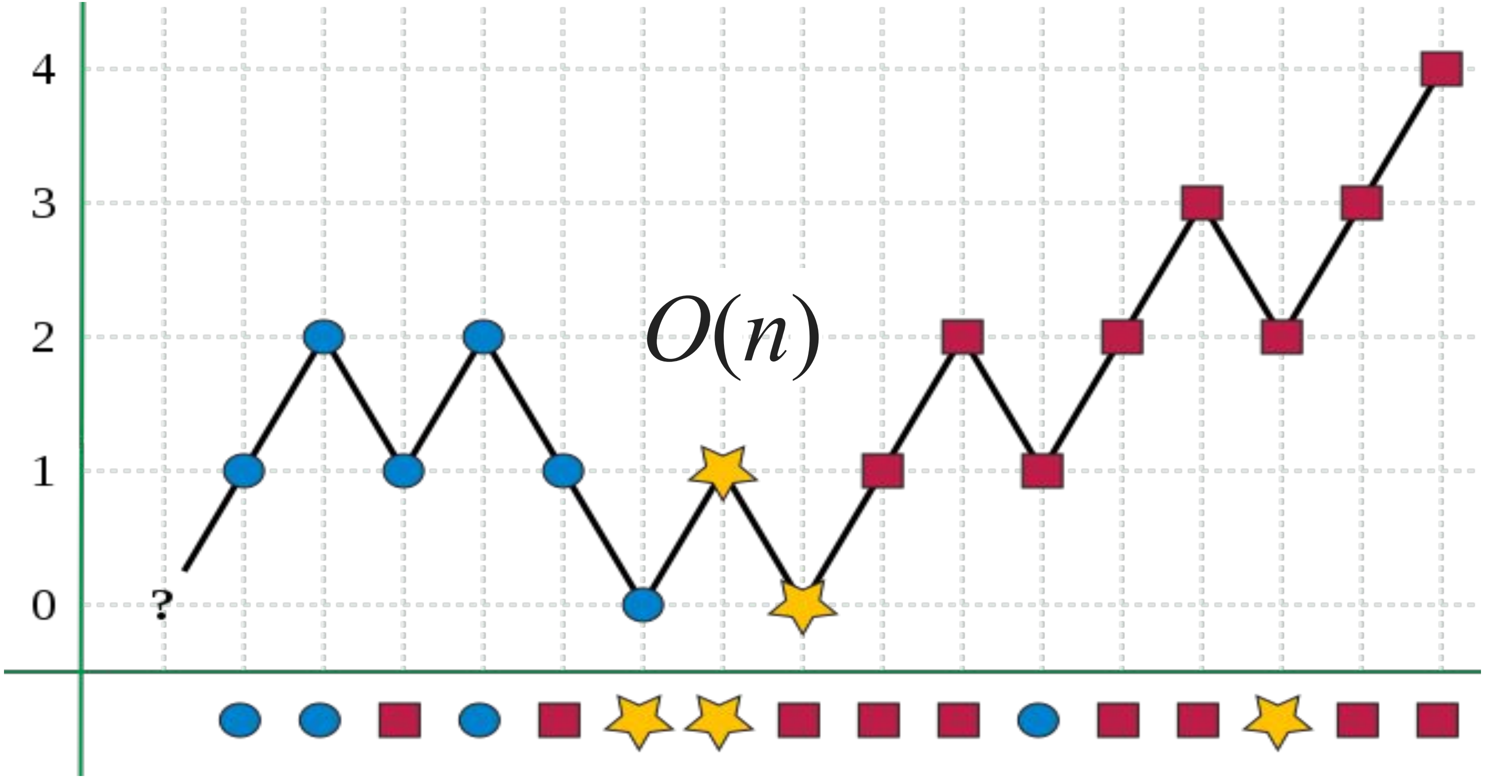
$O(nlogn)$



Calculating majority



Calculating majority



Induction

- *Start with $P(n)$*
- **Base case**: Show that the statement holds for $n = 0$ or n = 1.
- **Assumption** Assume that $P(k)$ holds
- **Inductive step**: Show that *if* $P(k)$ holds, then also $P(k + 1)$ holds

Interview Questions!!!

Interview Questions!!!

Interview Questions!!!

terview Questions!!!

Palindrome Permutation

Given a string, write a function to check if it is a permutation of a palindrome. A palindrome is a word or phrase that is the same forwards and backwards. A permutation is a rearrangement of letters. The palindrome does not need to be limited to just dictionary words.

EXAMPLE

Input: Tact Coa

Output: True (permutations: "taco cat", "atco cta"; etc.)

Palindrome Permutation

- Count each character (in a hash map)
- Check that each character has an even count
- Exactly one character can be odd

Smallest Difference

Given two arrays of integers, compute the pair of values (one value in each array) with the smallest (non-negative) difference. Return the difference.

EXAMPLE

Input: {1, 3, 15, 11, 2}, {23, 127, 235, 19, 8}

Output: 3. That is, the pair (11, 8).

Smallest Difference

Input: {1, 3, 15, 11, 2}, {23, 127, 235, 19, 8}

Sort the input:

{1, 2, 3, 11, 15}, {8, 19, 23, 127, 235}

Build Order

Build Order: You are given a list of projects and a list of dependencies (which is a list of pairs of projects, where the second project is dependent on the first project). All of a project's dependencies must be built before the project is. Find a build order that will allow the projects to be built. If there is no valid build order, return an error.

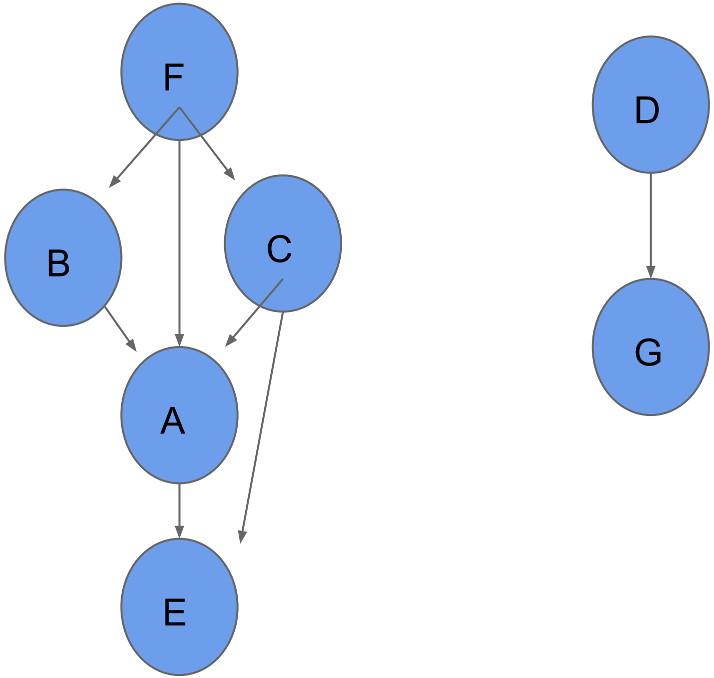
EXAMPLE

Input:

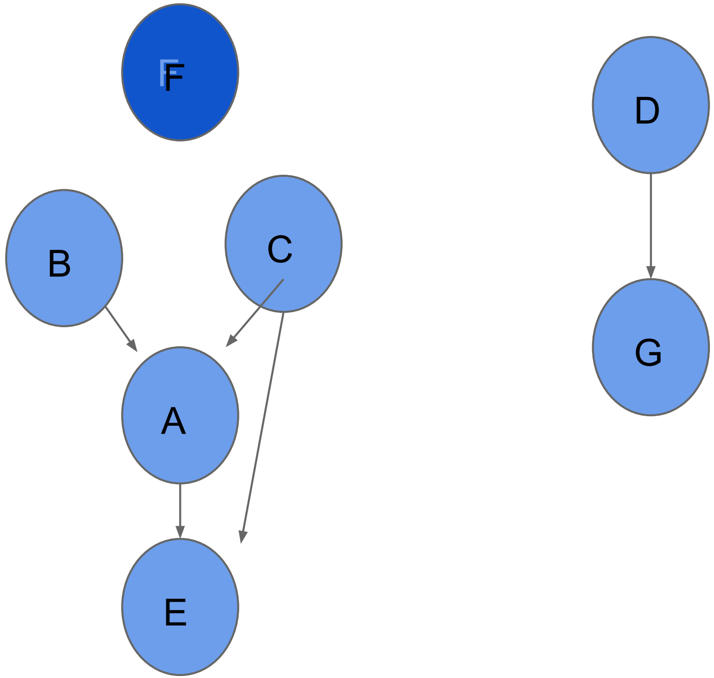
projects: a, b, c, d, e, f

dependencies: (a, e), (f, b), (b, a), (f, c), (d, g) (c, a) (f,a) Output: f, d, c, b, a, g, e

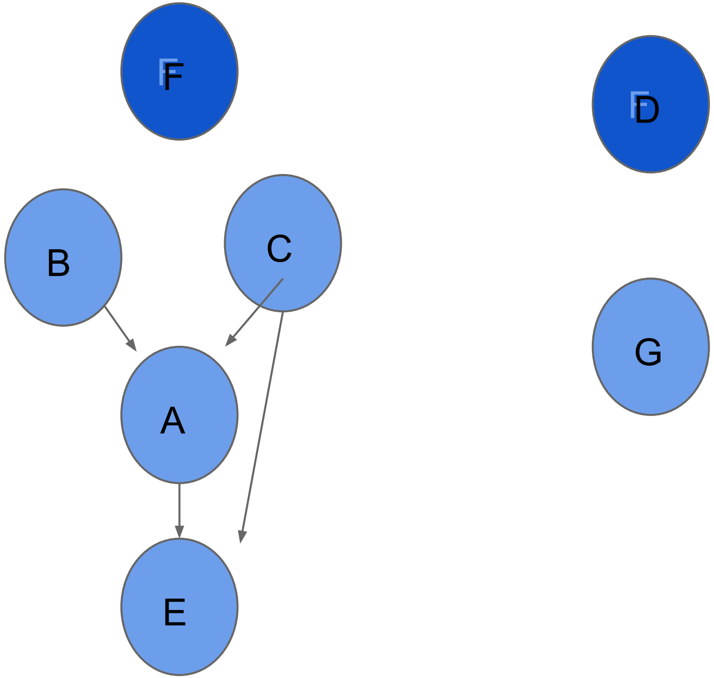
Build dependencies



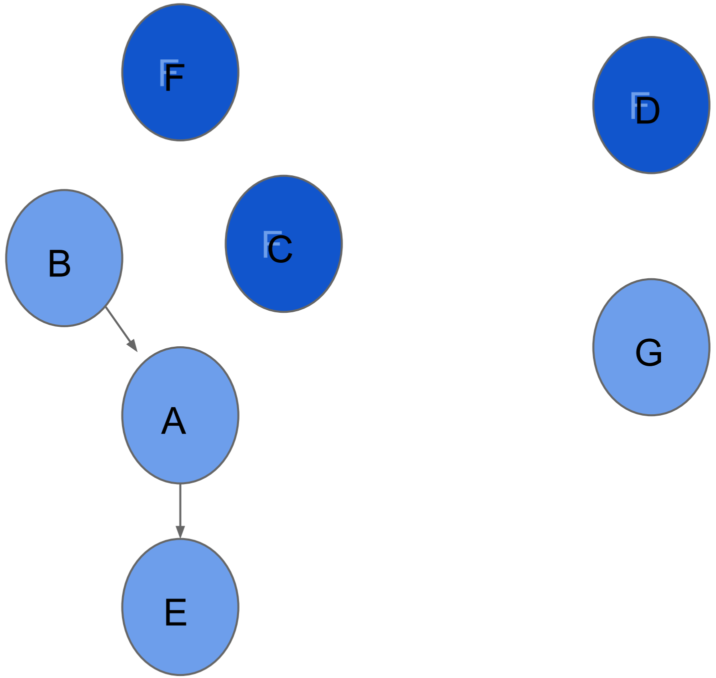
Build dependencies



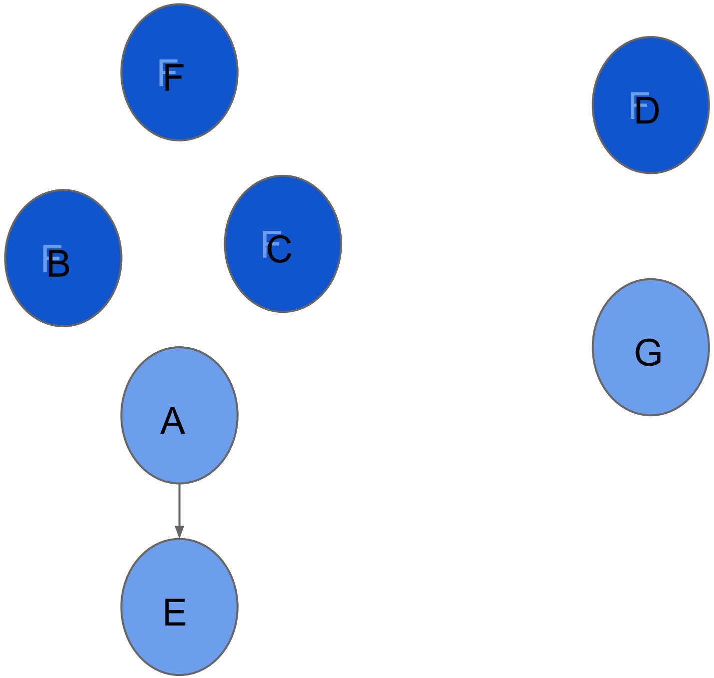
Build dependencies



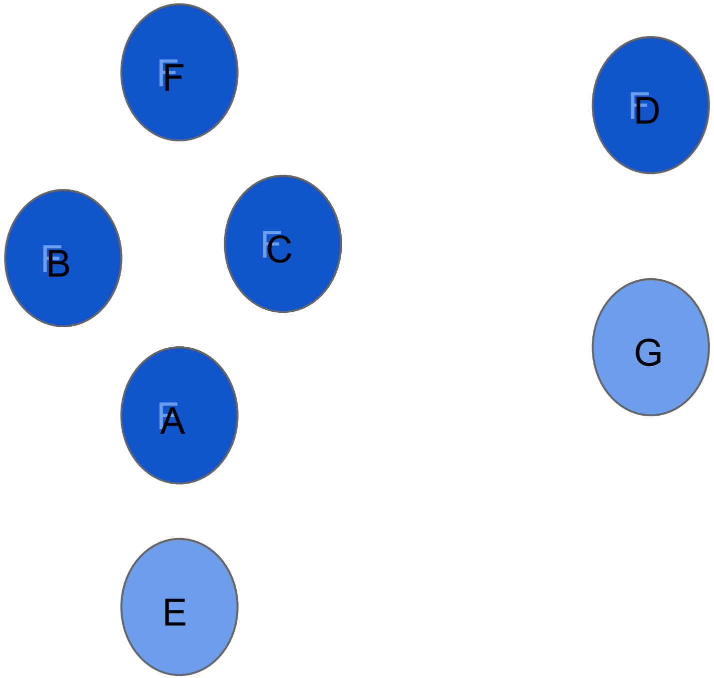
Build dependencies



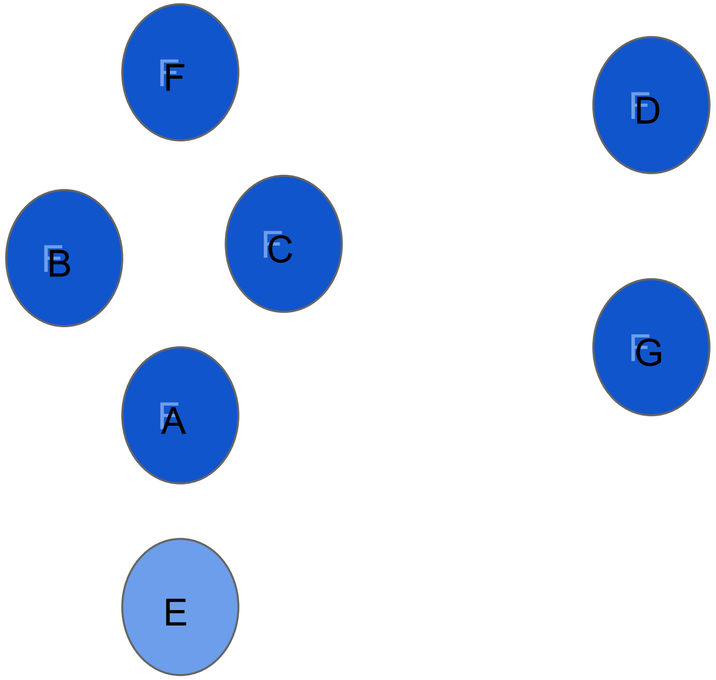
Build dependencies



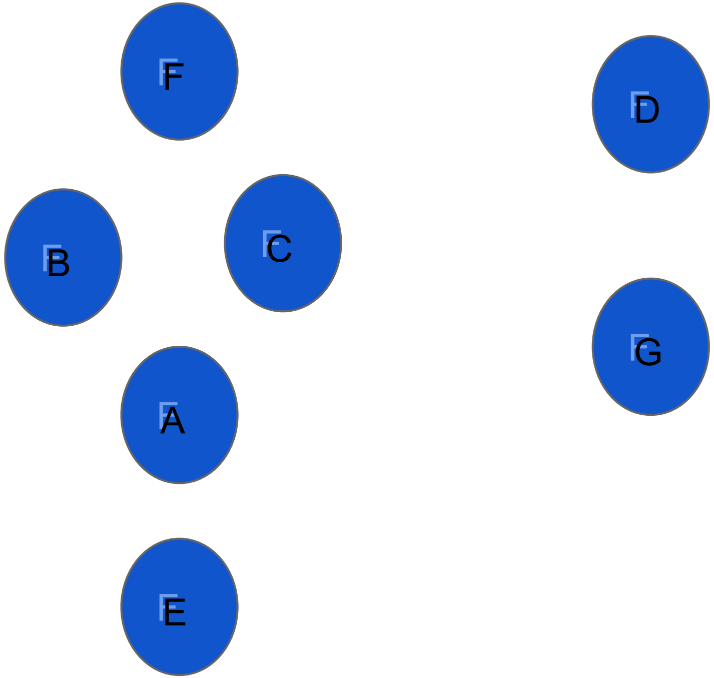
Build dependencies



Build dependencies

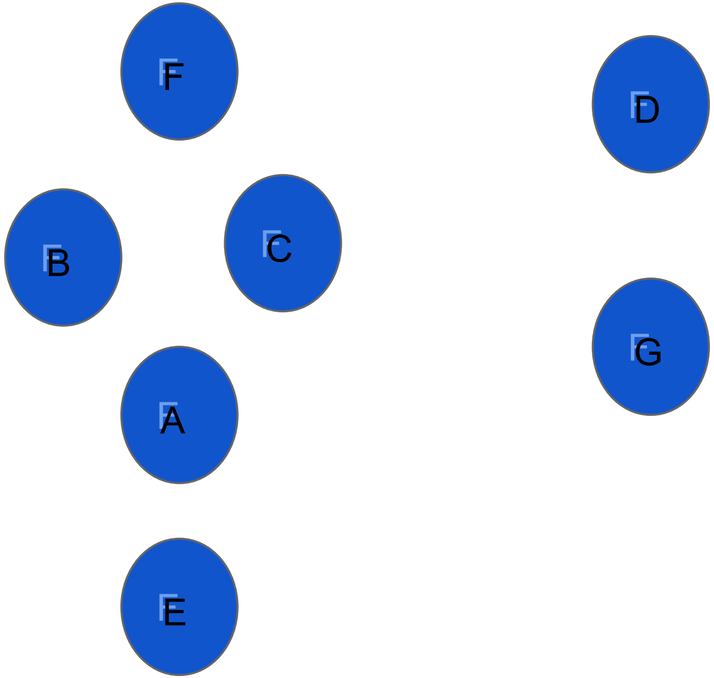


Build dependencies



Build dependencies

$O(P + D)$



CS180 Discussion

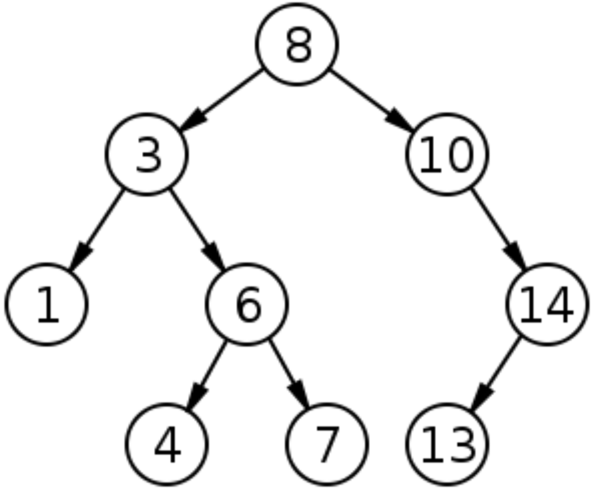
Week 2

Lecture Recap

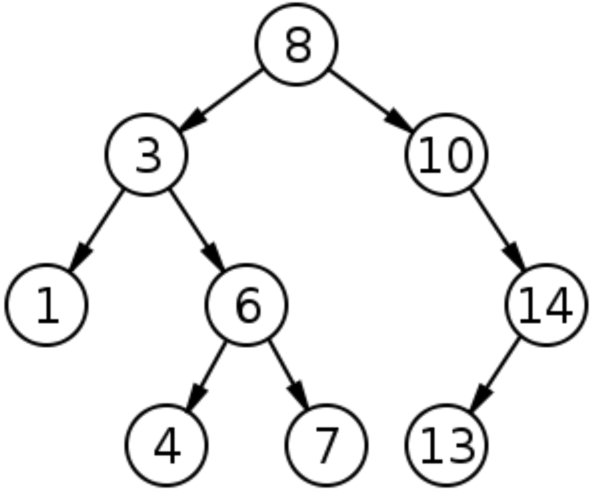
- Greedy algorithm
- Interval scheduling
- Intro to graphs
- BFS
- DFS
- ~~Graph coloring~~
- ~~Topological ordering~~

Binary Search Tree

A binary search tree is a rooted binary tree, whose internal nodes each store a key and each have two distinguished sub-trees, commonly denoted left and right. The tree additionally satisfies the binary search property, which states that the key in each node must be greater than or equal to any key stored in the left subtree, and less than or equal to any key stored in the right subtree

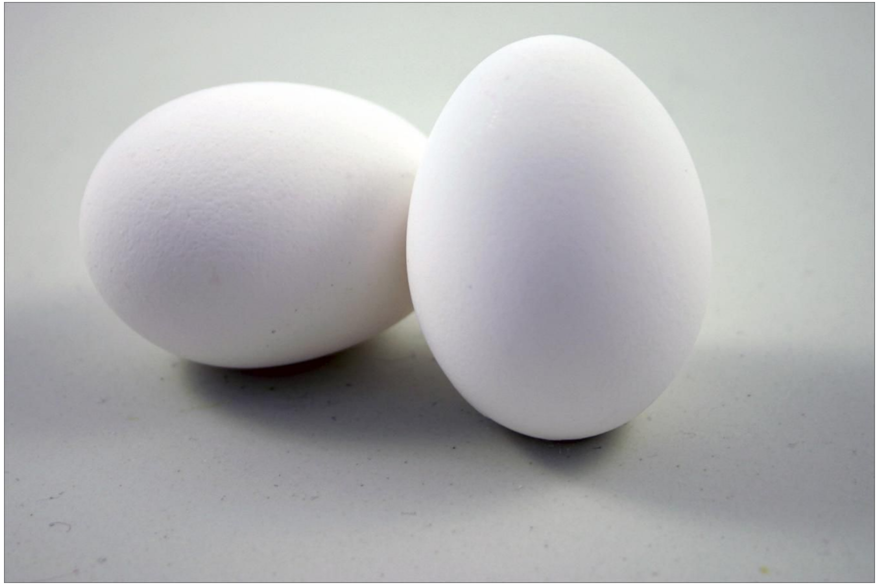


Question



HW 2 Questions

Minimization of Maximum Regret



Hospitals

There were m hospitals, each with a certain number of available positions for hiring residents. There were n medical students graduating in a given year, each interested in joining one of the hospitals. Each hospital had a ranking of the students in order of preference, and each student had a ranking of the hospitals in order of preference. We will assume that there were more students graduating than there were slots available in the m hospitals.

Ships

Given the schedule for each ship, find a truncation of each so that the condition continues to hold: no two ships are ever in the same port on the same day. Show that such a set of truncations can always be found, and give an algorithm to find them.

Example. Suppose we have two ships and two ports, and the “month” has four days. Suppose the first ship’s schedule is:
port P1; at sea; port P2; at sea
and the second ship’s schedule is:
at sea; port P1; at sea; port P2

TV Schedule

Suppose we have two television networks, whom we'll call A and B. There are n prime-time programming slots, and each network has n TV shows. Each network wants to devise a schedule—an assignment of each show to a distinct slot—so as to attract as much market share as possible.

Resolve this question by doing one of the following two things:

(a) give an algorithm that, for any set of TV shows and associated ratings, produces a stable pair of schedules; or

(b) give an example of a set of TV shows and associated ratings for which there is no stable pair of schedules.

terview Questions!!!

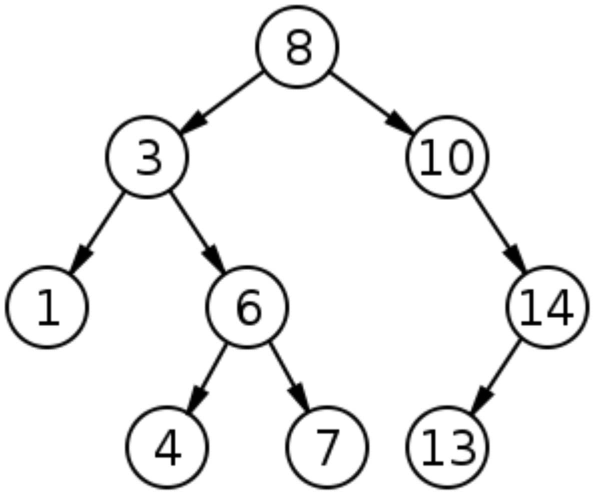
Interview Questions!!!

Interview Questions!!!

Interview Questions!!!

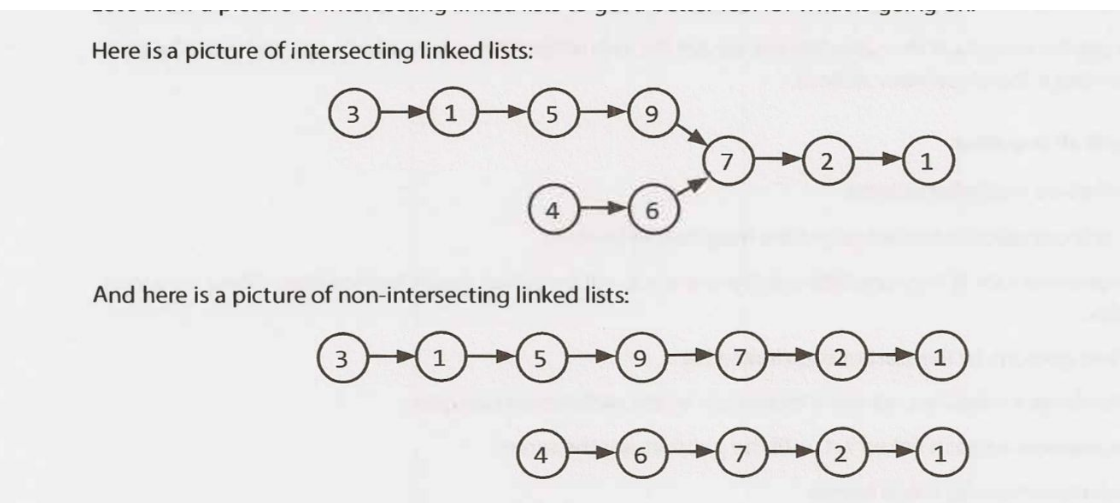
Validate BST

Implement a function to check if a binary tree is a binary search tree.



Intersection

intersection: Given two (singly) linked lists, determine if the two lists intersect. Return the intersecting node. Note that the intersection is defined based on reference, not value. That is, if the kth node of the first linked list is the exact same node (by reference) as the jth node of the second linked list, then they are intersecting.



Detecting Loops

Loop Detection: Given a circular linked list, implement an algorithm that returns the node at the beginning of the loop.

DEFINITION:

Circular linked list: A (corrupt) linked list in which a node's next pointer points to an earlier node, so as to make a loop in the linked list.

EXAMPLE

Input: A -> B -> C -> D -> E -> C [the same C as earlier] Output: C