NAME: ______________________________

COMPSCI 250
Introduction to Computation
Second Midterm Spring 2022

D. A. M. Barrington and K.Doney 7 April 2022

DIRECTIONS:

- Answer the problems on the exam pages.
- There are five problems on pages 2-8, some with multiple parts, for 100 total points plus 10 extra credit. Probable scale is somewhere around A=95, C=65, but will be determined after we grade the exam.
- Pages 9 and 10 (printed back to back) contains useful definitions and is given to you separately – do not put answers on it!
- If you need extra space use the back of a page.
- No books, notes, calculators, or collaboration.
- In case of a numerical answer, an arithmetic expression like “$2^{17} - 4$” need not be reduced to a single integer.

Question 1 (10): Recall that the Fibonacci function $F(n)$ is defined by the rules $F(0) = 0$, $F(1) = 1$, and for all n with $n > 1$, $F(n) = F(n-1) + F(n-2)$. In this problem you are going to solve a *similar*, but *different* recurrence. Here we have two base cases $h(0) = 1$ and $h(1) = 2$, and the general rule for positive naturals n is $h(n+1) = h(n) + 2h(n-1)$. Prove, by strong induction on all naturals n, that $h(n) = 2^n$. You will need two base cases.

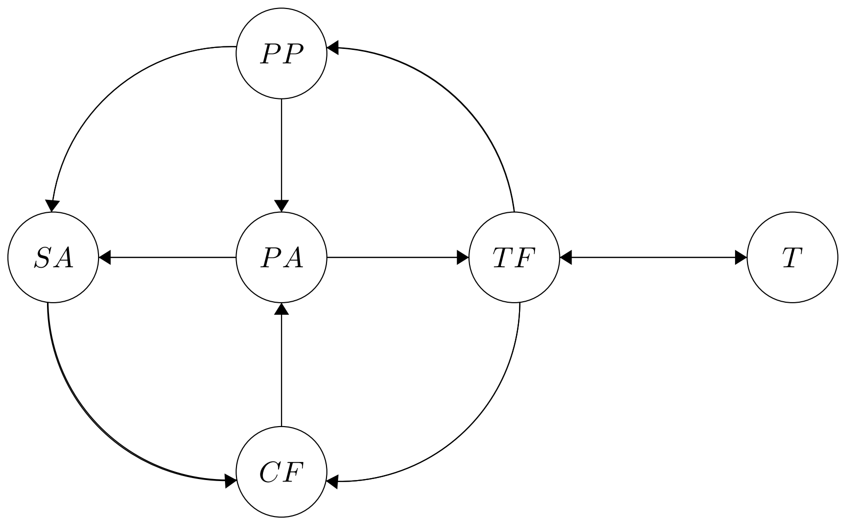
Question 2 (20+10): Let $\Sigma = \{a, b\}$, so that we are considering strings of a's and b's. A string is defined to have a **triple letter** if it contains either aaa or bbb as a substring, that is if it ever has the same letter three times in a row. For any natural, let $g(n)$ be the number of strings of length n that *do not* have a triple letter.

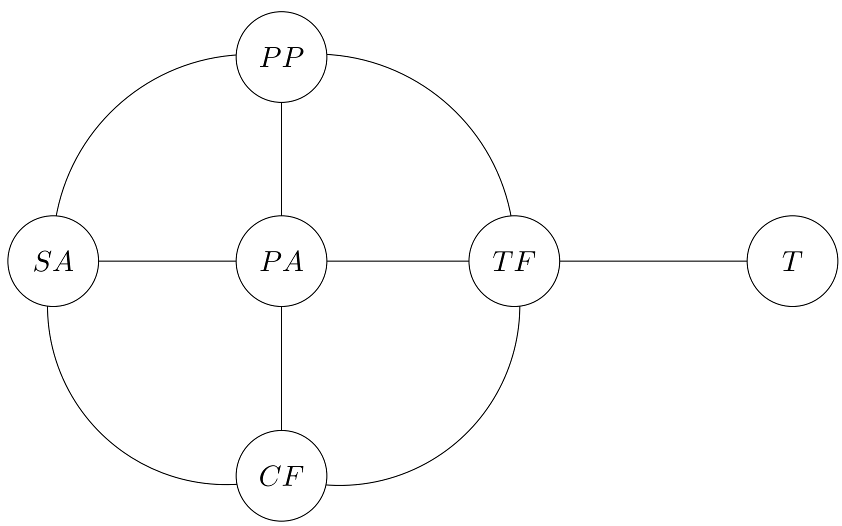
- (a, 10) Determine the values of $g(0)$, $g(1)$, $g(2)$, $g(3)$, and $g(4)$ by listing all the elements with no triple letter, of length at most 4. (You might want to compute $g(5)$ but it is not required. Show that $g(3) = g(2) + g(1)$ and $g(4) = g(3) + g(2)$. (You might also want to verify $g(5) = g(4) + g(3)$ but this is not required.)

- (b, 10) Assume that $g(n+1) = g(n) + g(n-1)$ for all n with $n > 1$. Prove, by strong induction on all positive naturals n, that $g(n) = 2F(n+1)$, where F is the ordinary Fibonacci sequence defined in Question 1. You will need two base cases, which you can get from part (a).

- (c. 10) Prove, for all naturals n with $n > 1$, that $g(n+1) = g(n) + g(n-1)$. (**Hint:** This problem does not necessarily require induction. If you have an arbitrary string of length $n+1$ with no triple letter, look at the case where the last two letters are different and the case where the last two letters are the same.)

Question 3 (20): Pictured here is a directed graph D with six nodes – note that there are edges both from TF to T and from T to TF. Also pictured is the undirected graph U with the same nodes, with undirected edges in place of each directed edge in D.



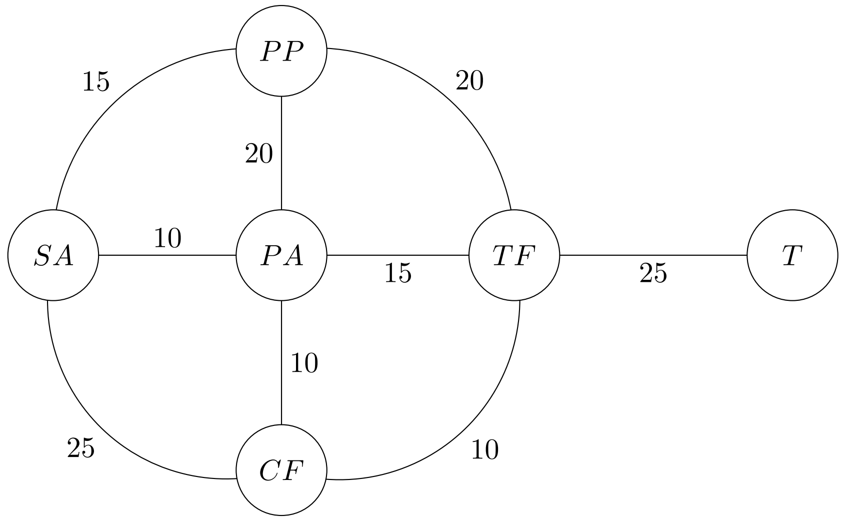


The two questions follow on the next page.

- (a, 10) Carry out a DFS search for the **directed** graph D starting with node CF. When two or more nodes need to come off the stack and they entered at the same time, take the one first that comes earlier alphabetically. Draw the DFS tree, indicating the non-tree edges, and classify each as a back, cross, or forward edge.

- (b, 10) Carry out a BFS search for the **undirected** graph U, starting with node CF. If two or more nodes need to come off the queue and they entered at the same time, take the one first that comes earlier alphabetically. Draw the BFS tree, indicating the non-tree edges.

Question 4 (30): Let G be the weighted undirected graph pictured here:



The six nodes in the graph represent locations in Rome, and you are currently at T (the train station) and you want to navigate from there to SA (the Castel Sant'Angelo near your hotel). The edges of the graph indicate walking time in minutes to go from one location to another – for example, it takes 25 minutes to walk from T to TF.

But there is a complication. Your companion insists that whenever we reach one of these locations, you have to stop to visit for the required time in the following table. (So, for example, traveling from T to TF will take you 40 minutes total, 25 for the walk and 15 to visit TF.)

- CF (Campo de Fiori): 25 minutes
- PA (Pantheon): 35 minutes
- PP (Piazza del Popolo): 10 minutes
- SA (Castel Sant'Angelo, your hotel): 10 minutes
- T (Termini, the train station): 0 minutes
- TF (Trevi Fountain): 15 minutes

Your goal is to reach the hotel with the minimum *total* time, and we'll eventually do this with an A^* search using the walking time as a heuristic. We want $h(x)$ for each node to be the walking time from the *goal*, SA, to x.

The two questions follow on the next page.

- (a, 15) Trace a uniform-cost search with start node SA and no goal node, using the walking times given on the graph, to find the value of $h(x)$ for every node x. Indicate which nodes are on the priority queue at each stage of the search.

- (b, 15) Conduct a complete A^* search of G with start node T and goal node SA, using the *total time* for each edge traversed. We determine the added time for each new edge by adding the walking time for that edge and the visit time for the destination. We use the values of heuristic function h from part (a). Indicate which nodes are on the priority queue at each stage of the search.

Question 5 (20): The following are ten true/false questions, with no explanation needed or wanted, no partial credit for wrong answers, and no penalty for guessing. Some of them refer to the scenarios of the other problems, and/or the entities defined on the supplemental sheet.

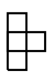
- (a) Let $\phi(x)$ be a predicate on the naturals. If $\phi(0)$ and $\phi(1)$ are true and we have $\forall x : \phi(x) \rightarrow (\phi(2x) \vee \phi(2x+1))$ being true, then $\forall x : \phi(x)$ is true.

- (b) Let $\phi(x)$ be a predicate on the naturals. If $\phi(0)$, $\phi(1)$ and $\phi(2)$ are true and we have $\forall x : \phi(x) \rightarrow (\phi(x-1) \rightarrow \phi(x+2))$ being true, then $\forall x : \phi(x)$ is true.

- (c) Let G be an arbitrary weighted graph with all non-negative weights. Let p_i be the optimal path weight from node i to the goal node g. Let e_i be the minimum number of steps to G from node i to the goal g. The heuristic $h(i) = p_i - e_i$ may fail to be an admissible heuristic for the A* algorithm.

- (d) In the undirected graph U for Question 3, there are exactly two articulation points.

- (e) If m and n are positive integers, then it is not possible to tile every $4m$ by $4n$ rectangle with T shaped tetrominoes.



- (f) If G is a game with a finite game tree, and every leaf is labeled with -1, 0, or 1, then the value of the game must be -1, 0, or 1.

- (g) Given the arithmetic expression "`+ + * 3 + + 5 7 * 12 6 2 * 7 + 14 1`" in prefix notation, the corresponding infix and postfix notation for the same expression are "`3*((5 + 7) + 12 * 6) + 2 + (7 * (14 + 1))`" and "`3 5 7 + 12 6 * + * 2 + 7 14 1 + * +`" respectively.

- (h) To prove a set with two operations form a semiring, it is sufficient to show that both the operators $+, \cdot$ are commutative and associative.

- (i) An arbitrary strongly connected directed graph G, with at least two nodes, must have at least one directed cycle.

- (j) An undirected graph G is bipartite if and only if there exist no cycles in G.

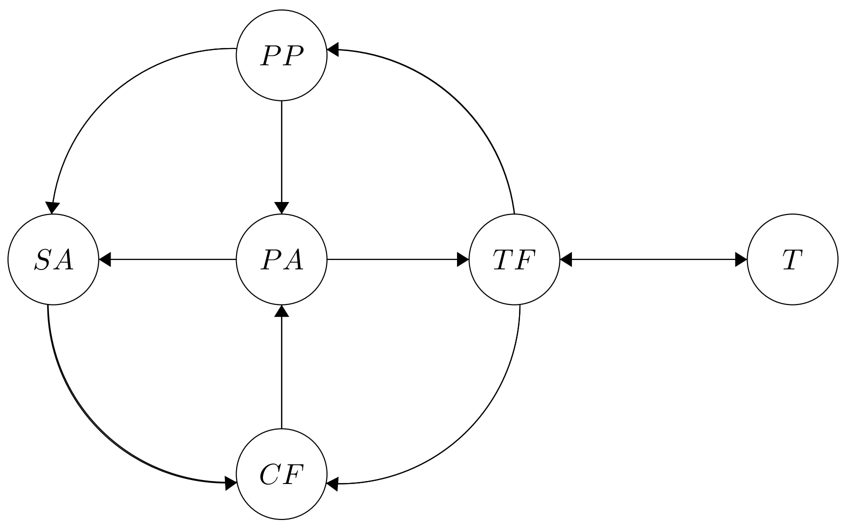
Supplemental Page for CS250 Midterm #2, 7 April 2022, Barrington/Doney

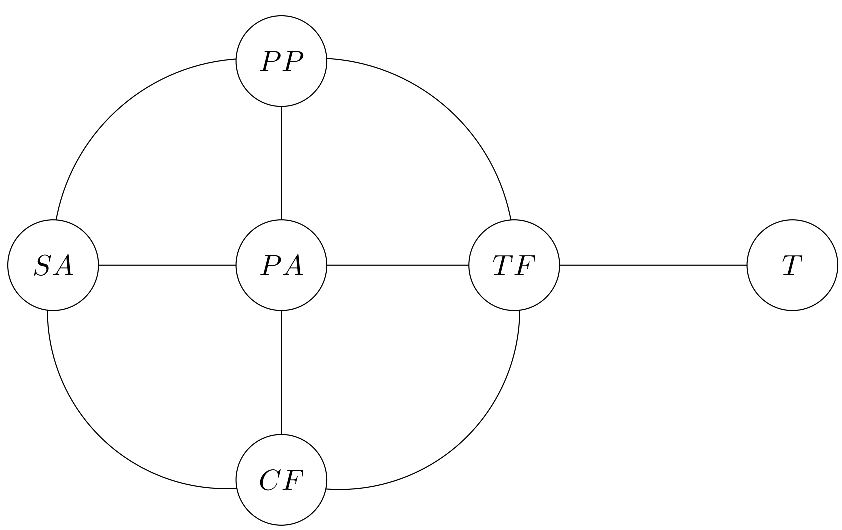
Questions 1 and 2 refer to the Fibonacci function, defined recursively by base cases $F(0) =$ and $F(1) = 1$, and the inductive case $F(n+1) = F(n) + F(n-1)$ for all positive naturals n.

Question 1 uses a different recursively defined function with base cases $h(0) = 1$ and $h(1) = 2$, and the inductive case $h(n+1) = h(n) + 2h(n-1)$ for all positive naturals n.

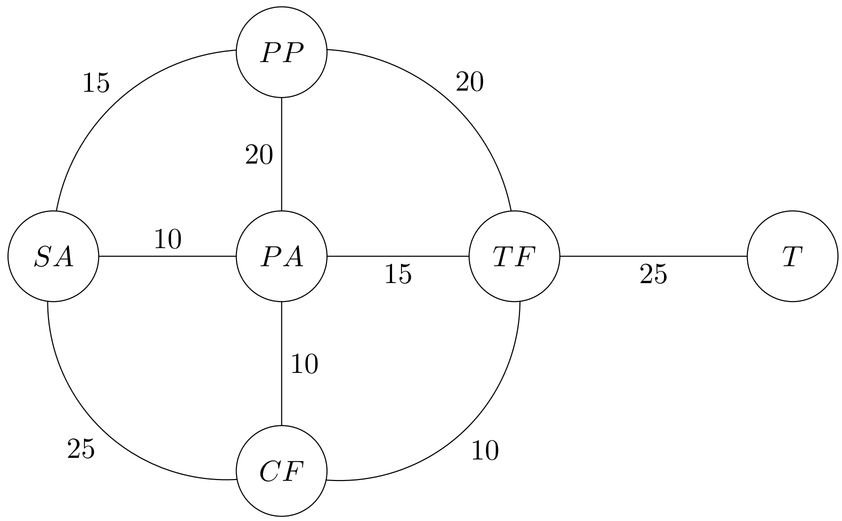
Question 2 defines a triple letter in a string over $\{a, b\}$, and defines $g(n)$ to be the number of strings of length n that *do not* have a triple letter.

Question 3 defines the following directed graph D and the undirected graph U, pictured below. Note that D has edges both from TF to T and from T to TF.





Question 5 deals with the weighted undirected graph pictured here. The weights define the walking time from one node to another.



Also given are the required visit times for each of the six nodes in the graph:

- CF (Campo de Fiori): 25 minutes
- PA (Pantheon): 35 minutes
- PP (Piazza del Popolo): 10 minutes
- SA (Castel Sant'Angelo, your hotel): 10 minutes
- T (Termini, the train station): 0 minutes
- TF (Trevi Fountain): 15 minutes