NAME: _____

COMPSCI 250
Introduction to Computation
SOLUTIONS to Second Midterm Spring 2023

D. A. M. Barrington and G. Parvini 25 April 2023

DIRECTIONS:

- Answer the problems on the exam pages.
- There are six problems on pages 2-10, 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.
- 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 (15): Recall that the Fibonacci function $F(n)$ is defined by the rules $F(0) = 0$, $F(1) = 1$, and for all n with $n > 1$, $F(n) = F(n-1) + F(n-2)$. Show that for all *positive* naturals n,

$$F(0)F(1) + F(1)F(2) + \ldots + F(2n-1)F(2n) = F(2n)^2.$$

Solution: Base case: n=1 : $F(0)F(1) + F(1)F(2) = 1 * 1 = F(2)^2$.
IH: Let's assume for some arbitrary k, $F(0)F(1)+F(1)F(2)+\ldots+F(2k-1)F(2k) = F(2k)^2$.
IS: We show the statement is true for $k+1$*:*
$F(0)F(1)+F(1)F(2)+\ldots+F(2k-1)F(2k)+F(2k)F(2k+1)+F(2k+1)F(2k+2) = F(2k+2)^2$.
By the IH: $F(0)F(1)+F(1)F(2)+\ldots+F(2k-1)F(2k)+F(2k)F(2k+1)+F(2k+1)F(2k+2) = F(2k)^2+F(2k)F(2k+1)+F(2k+1)F(2k+2) = F(2k)(F(2k)+F(2k+1))+F(2k+1)F(2k+2) =$ *(By Fibonacci Def)* $F(2k)F(2k+2)+F(2k+1)F(2k+2) = F(2k+2)(F(2k)+F(2k+1)) =$ *(By Fibonacci Def)* $F(2k+2)F(2k+2) = F(2k+2)^2$.

Question 2 (25): Prove the two questions using induction.

- (a, 15) In the town of Gigilis, every year, exactly one child is born. Once the child is born, we say they are one year old. Every year the age of a Gigilis child multiplies by two (i.e. they become 8 when a typical child in a typical town is 3). You are visiting the town of Gigilis, and they ask how old you are. They don't understand the usual numbers, so you should express your age as the summation of the ages of the people in their town. For example, if Aigili is 1, Cigili is 4, and Eigili is 16 and you are 21 you should tell them that you are Eigili + Cigili + Aigili years old. Prove by induction that for any positive natural $n > 0$, any age n can be expressed in this way, assuming that there are enough children with the specified ages. Note that you cannot use someone's name twice.

  The question asks if we can write any number as a summation of distinct powers of 2.
  Base Case: For $n = 1$, $1 = 2^0$.
  IH: We use Strong induction here: Let's assume for any number smaller or equal to n, we can write it as summation of distinct powers of 2. Now we show we can express n+1 as a summation of distinct powers of 2. If $n+1$ *is odd, then n is even and so we didn't use any* 2^0 *when we wanted to express n. So by adding* 2^0 *we find what we want. But if* $n+1$ *is even then n is odd and we cannot add* 2^0 *since it already exists. We know* $n+1$ *is even so we can write it as* $n+1 = 2k$ *for some natural k. Since* $0 \leq k \leq n$ *by strong induction hypothesis we know we can write k as summation of distinct powers of 2. We add one to each of these powers. The result is exactly twice of k and so equal to* $n+1$ *since we added one to all the powers they are all still distinct.*

- (b, 10, XC) The Gigilis children want to play a game. Each player chooses a binary string w, repeat it for i times (the concatenation of i copies of w when i is a non-negative integer), and the next player has to reverse it. For example, if the first player chooses $w = 001$ and $i = 4$, then will say "011011011011" and the next player has to say "110110110110". However, the children have a difficult time reversing a long string.

  Eigili, who recently turned 16 and started learning about induction, tells them they can reverse their string first and then repeat it i number of times instead. However, since Eigili learned the induction lesson recently, she cannot prove it and convince the others. Help Eigili by proving what she says is correct. In the other words prove by induction that for any string w and any natural i,

  $$(w^R)^i = (w^i)^R.$$

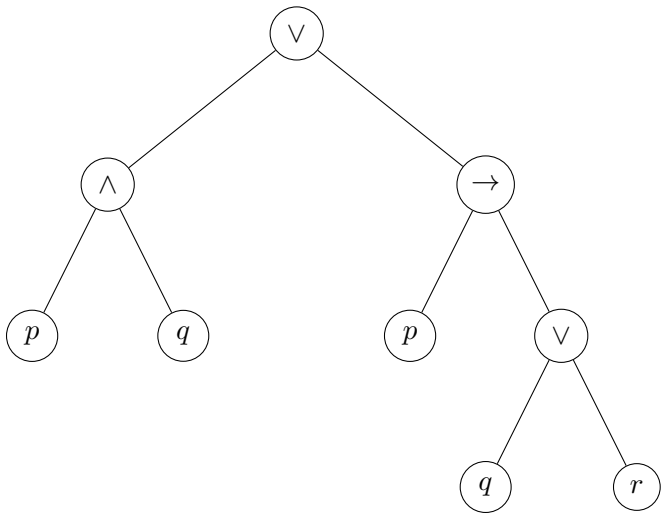
  (**Hint:** We have the definitions of reversal $\lambda^R = \lambda$ and $(wa)^R = aw^R$ and of exponentiation $w^0 = \lambda$ and $w^{i+1} = (w^i)w$. The rule $w(w^i) = w^{i+1}$ is not part of the definition, but we will let you use it without proof. Also, you may use the fact (proved in lecture) that for any strings u and v, $(uv)^R = v^R u^R$.)

  Lemma: For any string w and any natural i, $ww^i = w^{i+1}$.
  Base: $i = 0$, $w\lambda = w^1$.
  IH: $ww^i = w^{i=1}$,
  IG: $ww^{i+1} = w^{i+2}$.
  Proof: $w(w^{i+1})$ is $w(w^i w)$ by definition of exponentiation, which is $(ww^i)w$ by associativity, which is $w^{i+1}w$ by the IH, which is w^{i+2} by definition of associativity.

  Let w be an arbitrary string and use ordinary induction on all naturals i. Base case: $i = 0$, so $(w^R)^i$ and w^i are each λ by the definition of exponentiation, and then $(w^i)^R$ is also λ by the definition of reversal. IH: $(w^R)^i = (w^i)^R$ IG: $(w^R)^{i+1} = (w^{i+1})^R$ Proof of inductive step: $(w^R)^{i+1}$ is $(w^R)^i w^R$ by the definition of exponentiation, which is $(w^i)^R w^R$ by the IH, which is $(ww^i)^R$ by the rule $(uv)^= v^R u^R$ which we proved in lecture, which is $(w^{i+1})^R$ by the Lemma.

Question 3 (10): Answer these two questions briefly.

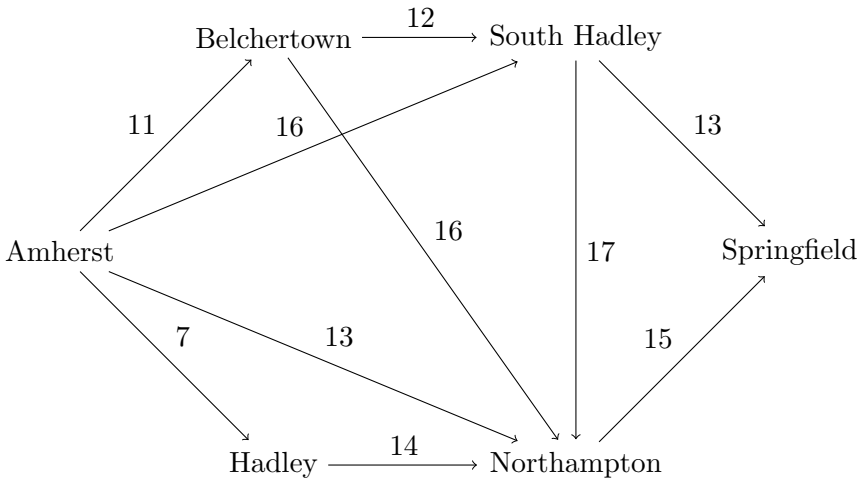
- (a, 5) Draw the parse tree for the formula $(p \wedge q) \vee (p \rightarrow (q \vee r))$ and find the postfix expression for this tree.
  The tree is pictured here:



  The postfix expression is "$pq \wedge pqr \vee \rightarrow \vee$".

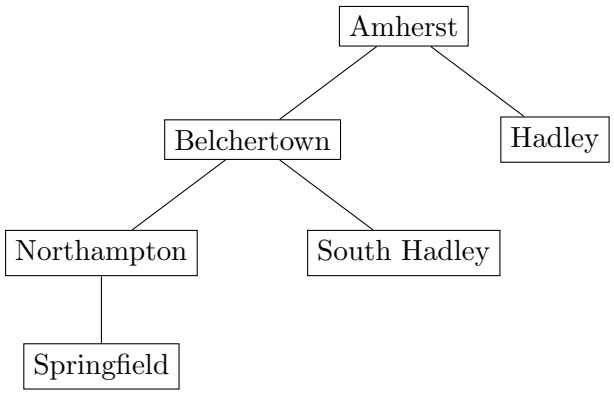
- (b, 5) We define the complement of an undirected graph G, as a graph $\overline{G}$ with the same vertex set and edge set as $\overline{E} = \{(x, y) | (x, y) \notin E\}$. How many edges exist in $\overline{G}$ when G has 8 nodes and 17 edges?
  There are $\binom{8}{2} = \frac{n(n-1)}{2} = 28$ *possible edges* 17 *of those possible edges,* $\overline{E}$ *contains the other* $28 - 17 = 11$.

Question 4 (20): Pictured here is a directed graph G with six nodes. Its undirected version U is pictured on a following page. The weights are not used in this question.



Answer the two following questions about the graphs G and U.

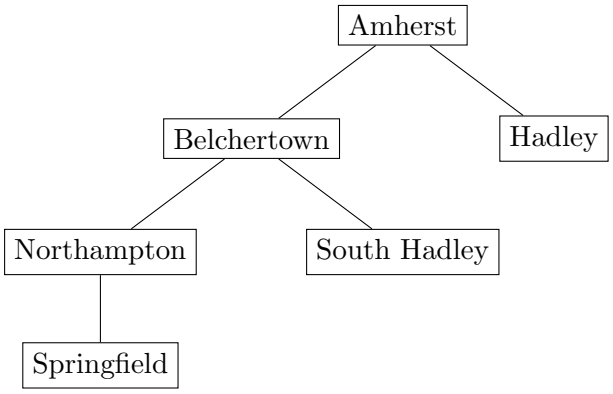
- (a, 10) Carry out a DFS search for the **directed** graph starting with node Amherst. When two or more nodes need to come off the stack, and they entered at the same time, take the one first that comes earlier alphabetically. Draw the DFS tree, indicating the non-tree edges, and classify each as a back, cross, or forward edge. Show each step of your stack, not only the final result.



  - *A begins on the stack.*
  - *A comes off, and B, H, N, and SH go on.*
  - *B comes off, N and SH go on, H, N, and SH remain on.*
  - *N comes off, S goes on, SH, H, N, and SH remain on.*
  - *S comes off, nothing goes on, SH, H, N, and SH remain on.*
  - *SH (from B) comes off, nothing goes on, H, N, and SH remain on.*
  - *H (from A) comes off, nothing goes on, N and SH remain on*
  - *The remaining nodes are discarded since they have been already processed.*

  The tree has A as the root. A has B and H as its children. B has N and SH as its children. N has S as a child. The other three nodes are leaves. There are five non-tree edges. (A, N) and (A, SH) are forward edges. (H, N), (SH, N), and (SH, S) are cross edges. There are no back edges, which makes sense as this directed graph is acyclic.

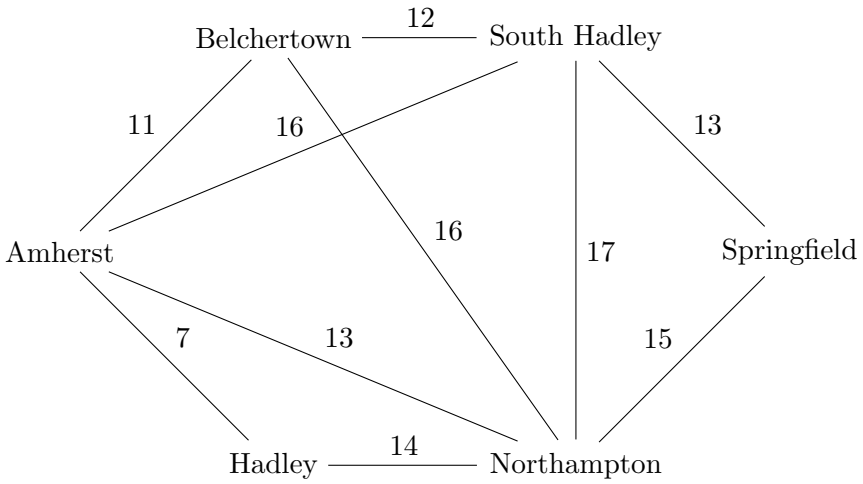
- (b, 10) Ignore the edges' directions and conduct a BFS search for the *undirected* graph U, with Amherst as the start node. If two or more nodes need to come off the queue and they entered at the same time, take the one first that comes earlier alphabetically. Draw the BFS tree, indicating the non-tree edges. Show each step of your work, not only the final result.



- *A begins on the queue.*
- *A comes off, and B, H, N, and SH go on.*
- *B comes off, H, N, and SH remain, and N and SH go on.*
- *H comes off, N, SH, N, and SH remain, and N goes on.*
- *N comes off, SH, N, SH, and N remain, and SH and S go on.*
- *SH comes off, N, H, SH, N, SH, and S remain, and S goes on.*
- *N, H, SH, N, and SH are all discarded, leaving only two copies of S on the queue.*
- *S (from N) comes off, nothing goes on, and only S remains.*
- *S is discarded and the queue is empty.*

The BFS tree has root A. A has four children B, H, N, and SH. S is a child of N. The five non-tree edges are (B, N), (B, SH),*(H, N),(N, SH),and(SH, S).*

Question 5 (20): Let U be the weighted undirected graph made from G, pictured here:



The six nodes in the graph represent locations in Massachusetts, and you are currently at Amherst, and you want to navigate from there to Springfield. The edges of the graph indicate traveling distance in miles to go from one location to another – for example, it takes 13 miles to drive directly from Amherst to Northampton.

Your goal is to reach Springfield with the minimum *total* distance.

The two questions follow on the next page.

- (a, 10) Trace a uniform-cost search with start node Amherst and no goal node, using the distances in miles given on the graph, to find the value of $d(x)$ for every node x. Indicate which nodes are on the priority queue at each stage of the search.
  - *Time* 0*:* $A0$*,* A *explored*
  - *Time* 1*:* $H7$*,* $B11$*,* $N13$*,* $SH16$*,* H *explored*
  - *Time* 2*:* $B11$*,* $N13$*,* $SH16$*,* $N21$*,* B *explored*
  - *Time* 3*:* $N13$*,* $SH16$*,* $N21$*,* $SH23$*,* N *explored*
  - *Time* 4*:* $SH16$*,* $N21$*,* $SH23$*,* $S28$*,* $S29$*,* $SH30$*,* SH *explored*
  - *Time* 5*:* $N21$ *discarded*
  - *Time* 6*:* $SH23$ *discarded*
  - *Time* 7*:* $S28$*,* $S29$*,* $SH30$*,* S *explored and we have found Springfield at distance 28.*

  Most people did well on this, but I gave a lot of 9/10 scores for small errors in the sequence of events. The most common was "discarding early". In our version of UCS, the node from any edge from exploring any node not yet closed has to be put onto the PQ, and should stay there until it emerges at its turn. So, for example, you should not discard the (N, 27) node before (S, 16) has come off of the PQ and been processed. If I could not tell when you had made the discard, I generally took a point off. Some people were essentially using the Dijkstra algorithm rather than our generic version of UCS, as they were comparing any new node with any existing node in the PQ for the same city. I generally took one or two points for this. I also took three points if you wound up with any answer other than 28 for the distance to Springfield.

- (b, 10) In the undirected graph U we define a heuristic function h for the goal node Springfield. For every node x in the graph, $h(x)$ is 11 times the minimum number of edges in any path from x to Springfield. Trace an A^* search with start node Amherst and goal node Springfield, using h as the heuristic function. (You may assume without proof that h is an admissible and consistent heuristic for this search.) Indicate which nodes are on the priority queue at each stage of the search.

Heuristic values: $h(S) = 0$*,* $h(SH) = h(N) = 11$*,* $h(A) = h(B) = h(H) = 22$

- *Time* 0*:* $(A, 0 + 22)$*, A explored*
- *Time* 1*:* $(N, 13 + 11)$*,* $(H, 7 + 22)$*,* $(SH, 16 + 11)$*,* $(B, 11 + 22)$*, N explored*
- *Time* 2*:* $(H, 7+22)$*,* $(SH, 16+11)$*,* $(B, 11+22)$*,* $(S, 28+0)$*,* $(SH, 30+11)$*,* $(H, 27+22)$*,* $B(29 + 22)$*, SH explored*
- *Time* 3*:* $(H, 7 + 22)$*,* $(B, 11 + 22)$*,* $(S, 28 + 0)$*,* $(SH, 30 + 11)$*,* $(H, 27 + 22)$*,* $B(29 + 22)$*,* $(SP29 + 0)$*, S explored and we have found Springfield at distance 28.*

We were able to determine the shortest past without needing to explore Hadley or Belchertown, making our search significantly shorter.

I took off one or two points for not putting all the explored nodes on the PQ. Many did not realize that when you have a node in the PQ, the algorithm does not know what value it has, so that if it finds another node for the same city it has to enqueue it. (The comparison will happen within the PQ, when it is time to take one of the nodes off.) I made a pretty serious deduction if I couldn't see on the PQ where you compared the path A-SH-S of length 29 with the path A-N-S of length 28. In an A search, you can't tell when you have found the right path until the goal node has come off of the PQ. Only then you know that any other candidate paths don't matter.*

There was a common mistake where people managed to count the heuristic for A-N-S twice, saying that the optimal math was 39 rather that 28. You should know that the optimal path is just the same as the one for the UCS.

Question 6 (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 $\phi(1)$ and $\phi(2)$ are true and we have $\forall x : \phi(x) \to (\phi(2x) \vee \phi(2x+1))$ being true, then $\forall x : \phi(x)$ is true.
  FALSE. It is possible that $\phi(3)$ is not true.

- (b) The proof that strong induction is valid uses ordinary induction.
  TRUE.

- (c) To compute the heuristic values in Question 5(b), we should use the breadth-first search from Question 4(b), multiplying the depth of each node by 11.
  FALSE. That search would be great if it were finding the number of edges to Springfield, but instead it finds the number of edges to Amherst.

- (d) Let H be an undirected graph, and let us conduct a DFS from some node. If the resulting DFS tree contains at least one back edge, then there exists a cycle in H.
  TRUE.

- (e) If T is any rooted tree with n nodes and e edges, then it is possible that $n = e + 2$.
  FALSE. The correct formula is $n = e + 1$.

- (f) The path relation in an undirected graph is always an equivalence relation.
  TRUE. It is reflexive, symmetric and transitive.

- (g) The path relation in a directed graph is always a partial order relation.
  FALSE. It is not necessarily anti-symmetric.

- (h) The number of subsets of size 2 in a set S with n elements is $\frac{n^2}{2}$.
  FALSE. It is $\frac{n(n+1}{2}$.

- (i) Let $P(x)$ be a property of **real** numbers. If we prove that $P(0)$ is true, and we prove that $\forall x : P(x) \to P(x+1)$, we may conclude that $P(x)$ is true for all real numbers x.
  FALSE. We have no method to use induction for real numbers.

- (j) The relation $D(x, y)$, in a rooted tree, is defined so that $D(x, y)$ is true if and only if x is y's descendant. Then we can show that $D(x, y)$ is an equivalence relation.
  FALSE. It is not symmetric.