NAME: ____________________

SPIRE ID: ____________________

COMPSCI 250
Introduction to Computation
SOLUTIONS to Second Midterm Fall 2023

D. A. M. Barrington and G. Parvini 9 November 2023

DIRECTIONS:

- Answer the problems on the exam pages.
- There are four problems on pages 2-8, some with multiple parts, for 100 total points plus 5 extra credit. Final scale will be determined after the exam.
- Page 9 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 –both sides are scanned.
- If absolutely necessary, you may attach additional pages to be graded, but this is a big nuisance to us.
- No books, notes, calculators, or collaboration.
- Scrap paper during the exam may be used, if it starts out blank. Please transfer any answers to the exam pages.
- You may say "pass" for any question or lettered part of a question, except for the true/false and the extra credit question, and receive 20% of the points.

1	/10
2	/15
3	/15
4	/40+5
5	/20
Total	/100+5

Question 1 (10): (Induction 1) Define a sequence of integers by the rule $a_0 = 7$ and, for any natural n, $a_{n+1} = a_n + n - 3$.

Prove, by induction, that for any natural n, $a_n = \frac{1}{2}(n^2 - 7n + 14)$.

(**Hint:** This can be done with ordinary induction.)

Base Case: a_0 is given as 7, $P(0)$ says that it is $\frac{1}{2}(0^2 - 7 \cdot 0 + 14) = 7$. IH: $a_n = \frac{1}{2}(n^2 - 7n + 14)$. IG: $a_{n+1} = \frac{1}{2}((n+1)^2 - 7(n+1) + 14)$. IS: We are given that $a_{n+1} = a_n + n - 3$. Applying the IH, $a_{n+1} = \frac{1}{2}(n^2 - 7n + 14 + 2n - 6)$. By algebra, we get $a_{n+1} = \frac{1}{2}(n^2 - 5n + 8)$, while the IG is $a_{n+1} = \frac{1}{2}(n^2 + 2n + 1 - 7n - 7 + 14)$ which is $\frac{1}{2}(n^2 - 5n + 8)$ as well, satisfying the inductive step and completing the induction.

Question 2 (15): (Induction 2) Prove, for any natural n, that the number $2^n 3^n$ divides the number $(3n)!$ (the factorial of $3n$).

Hint: Remember that $0! = 1$. You will probably want to compute the number $(3n+3)!/(3n)!$. Ordinary induction is sufficient.

Base Case: For $n = 0$, $2^0 3^0 = 1$ and $(3 \cdot 0)! = 0! = 1$. IH: $2^n 3^n$ divides $(3n)!$, so there exists some natural x such that $x 2^n 3^n = (3n)!$. IG $2^{n+1} 3^{n+1}$ divides $(3n+3)!$, so there exists some natural y such that $2^{n+1} 3^{n+1} = y(3n+3)!$. For the inductive step, we first write the left-hand side $2^{n+1} 3^{n+1} = 6 \cdot 2^n 3^n$, then write the right-hand side as $(3n+3)! = (3n)!(3n+1)(3n+2)(3n+3) = x 2^n 3^n (3n+1)(3n+2)(3n+3)$. Since $3n+3$ is divisible by 3, and either $3n+1$ or $3n+2$ is even, $(3n+1)(3n+2)(3n+3)$ is divisible by 6, that is, it is $6z$ for some natural z. We can thus rewrite the right-hand side as $xz 2^{n+1} 3^{n+1}$, and we can complete the inductive step by setting y to be xz.

Question 3 (15) (Induction 3) :

Let $\Sigma = \{a, b, c\}$. A **palindrome** is a string w whose reversal w^r is itself. For any natural n, let $pal(n)$ be the number of palindromes of length n over Σ.

Prove by induction for all *positive* naturals n that $pal(n) = 3^{(n+1)/2}$, where "/" here is floor division or Python "//". (So we are proving that $pal(n) = 3^{n/2}$ if n is even, and $pal(n) = 3^{(n+1)/2}$ if n is odd.)

(**Hint**: Determine the value of $pal(n+2)$ in terms of $pal(n)$, and complete the proof with either odd-even induction or strong induction.)

If w is any palindrome of length n, there are three palindromes awa, bwb, and cwc, each of length $n+2$ that are associated with it. Furthermore, every palindrome of length $n+2$ must be associated in this way with some palindrome of length n. Since there are exactly three palindromes of length $n+2$ for every palindrome of length n, we conclude that for any n, $pal(n+2) = 3pal(n)$.

For the even-odd induction, we first prove our statement $P(n)$ for all even n. For the base case, note that there is exactly one palindrome of length 0, so $pal(0) = 1$, and the formula gives us $3^{0/2} = 1$. Since $3^{(n+2)/2} = 3 \cdot 3^{n/2}$, our observation that $pal(n+2) = 3pal(n)$ completes the inductive step. The proof of $P(n)$ for all odd n is identical, except that our base case is $pal(1) = 3$ (which is true because a, b, and c are exactly the palindromes of length 1) and we must check that $3^{(n+2)/2} = 3 \cdot 3^{n/2}$ is still true for odd n and the divisions is floor division.

For the strong induction, we let $Q(n)$ be $(i \leq n \rightarrow P(i)$, prove $P(0)$ and $P(1)$ as above in the even-odd induction, and note that for $n \geq 1$, $Q(n+1)$ follows from $Q(n)$ and our observation above that $P(n-1) \rightarrow P(n+1)$.

Question 4 (40+5): (Searches and Search Trees) This problem concerns an undirected graph U and a directed graph D, each of which has nine nodes and eleven edges. Both are copied on the supplemental sheet. We begin with U:

```
(a) =================== (b)
 |                       |
 |                       |
(c) =========== (d) === (e)
 |               |       |
 |               |       |
(f) === (g) === (h) === (i)
```

- (a, 10) Carry out a DFS search for the undirected graph U, beginning with node c and with no goal node, indicating which nodes go on and off of the open list in what order. When two or more nodes need to come off the stack, and they entered at the same time, take the one that comes earlier alphabetically.

  Node c starts on the stack.

  c comes off, a, d, and f go on.

  a comes off, b goes on, d and f remain.

  b comes off, e goes on , d and f remain.

  e comes off, d and e go on, d and f remain.

  d comes off, c is seen for a back edge , h goes on, d and f remain.

  h comes off, g and i go on, d and f remain.

  g comes off, f goes on, i, d, and f remain.

  f comes off, c is seen for a back edge, i, d, and f remain.

  i comes off, e is seen for a back edge, and d and f remain.

  d and f are discarded. Draw the DFS tree, indicating the tree edges and the back edges.

  The tree consists mostly of a single path of tree edges, c to a to b to e to d, to h to g to f, with another tree edge from h to i, and back edges d to c, f to c, and i to e.

- (b, 10) Carry out a BFS search for the undirected graph U, beginning with node c and with no goal node, indicating which nodes go on and off of the open list in what order. When two or more nodes need to come off the queue, and they entered at the same time, take the one that comes earlier alphabetically.

  Node c starts on the queue.

  c comes off, a, d, and f go on.

  a comes off, b goes on after d and f.

  d comes off, e and h go on after f and b.

  f comes off, g goes on after b, e, and h.

  b comes off, e goes on after e, h, and g.

  e comes off, i goes on after h, g, and e

  h comes off, g and i go on after g, e, and i

  g comes off, h is seen as a non-tree edge, and e, i, g, and i remain

e **is seen as a non-tree edge to** b**, and** i**,** g**, and** i **remain**

i **comes off,** h **is seen as a non-tree edge, and** g **and** e **remain**

g **and** e **are discarded**

Draw the BFS tree, indicating the tree edges and the non-tree edges.

c **has parents** a**,** d**, and** f

a **has** b

d **has** e **and** h

f **has** g

b **has a non-tree edge to** e

e **has** i

h **has non-tree edges to** g **and** i

Here is the directed graph D. It is obtained from U by assigning exactly one direction for each of the edges in U. The caret symbols under nodes a, b, and e indicate directed edges upward to those nodes. The v symbols indicate directed edges downward to c and g.

```
(a) <================== (b)
 ^                       ^
 |                       |
(c) ==========> (d) <== (e)
 |               |       ^
 v               v       |
(f) <== (g) <== (h) ==> (i)
```

- (c, 15) Conduct a DFS of the directed graph D, beginning with node c and with no goal node, indicating which nodes go on and off of the open list in what order. Again, nodes that entered the stack at the same time come off in alphabetical order.

  Node c **starts on the stack.**

  c **comes off,** a**,** d**, and** f **go on**

  a **comes off,** d **and** f **remain**

  d **comes off,** h **goes on,** f **remains**

  h **comes off,** g **and** i **go on,** f **remains**

  g **comes off,** f **goes on,** i **and** f **remain**

  f **comes off,** i **and** f **remain**

  i **comes off,** e **goes on,** f **remains**

  e **comes off,** b **goes on,** d **is seen as a back edge,** f **remains**

  b **comes off,** a **is seen as a cross edge,** f **remains**

  f **is seen as a forward edge**

  Draw the DFS tree, indicating each edge as a tree edge, back edge, cross edge, or forward edge.

  c **has parents** a **and** d

  a **is a leaf**

  d **has parent** h

  h **has parents** g **and** i

g **has parent** f**, which is a leaf**

i **has parent** e

e **has parent** b**, which is a leaf**

We have non-tree edges (b, a) **(cross),** (e, d) **(back), and** (c, f) **(forward).**

- (d, 5) The **strongly connected component** of a node u in a directed graph is the set of nodes v such that there are paths both from u to v and from v to u. (Thus the strongly connected components are the equivalence classes of the equivalence relation $P(u, v) \wedge P(v, u)$.) Determine *all* the strongly connected components in the directed graph D. Justify your answers, possibly referring to your answer in part (c).

  The back edge from e **to** d **creates a directed cycle** $\{d, e, h, i\}$**, so these four nodes are all in the same component. The other five nodes are each in their own**

  component – c **has no edge in,** a **and** f **each have no edge out, and**

  b **and** g **each have one edge in and one edge out, but the edge out has no edges out of its own.**

- (e, 5XC) Conduct a BFS of the directed graph D, also beginning with node c. In this case it is sufficient to draw the tree with its tree and non-tree edges – we don't need the sequence of events.

  Node c **starts on the queue.**

  c **comes off,** a**,** d**, and** f **go on**

  a **comes off,** d **and** f **remain**

  d **comes off,** h **goes on after** f

  f **comes off,** h **remains**

  h **comes off,** g **and** i **go on**

  g **comes off,** f **is seen as a non-tree edge**

  i **comes off,** e **goes on**

  e **comes off,** d **is seen as a non-tree edge,** b **goes on**

  b **comes off,** a **is seen as a non-tree edge**

  For the tree:

  c **has parents** a**,** d**, and** f

  a **is a leaf**

  d **has parent** h

f **is a leaf**
h **has parents** g **and** i
g **is a leaf**
i **has parent** e
e **has parent** b**, which is a leaf**
We have non-tree edges (b, a)**,** (e, d)**, and** (g, f)

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.

- (a) An undirected graph with n nodes may have any integer number of edges from 0 through $(n^2 - n)/2$.

  TRUE. The maximum number is the set of all pairs (i, j) with $i < j$.

- (b) Let G be a directed graph such that every node has at least one outgoing edge. Then G must have at least one directed cycle, so it is not acyclic.

  TRUE. If we start at any node, and follow a path by taking new nodes, we must eventually revisit a node.

- (c) If n is a natural, and we know that $n\%5 = 4$ and $n\%7 = 2$, we may conclude that $n\%35 = 9$. (Here the % operator is the modular division operator, which in this case is the same in either Java or Python.)

  TRUE. The Chinese Remainder Theorem tells us, since 5 and 7 are relative prime, that this pair of congruences is equivalent to a single congruence modulo 35. Since 9 is both congruent to 4 modulo 5 and congruent to 2 modulo 7, it is the correct remainder.

- (d) If a, b, c, d, and e are any five strings, then the strings $a(bc)^R(de)^R$ and $ae^Rd^Rc^Rb^R$ are the same. (Here a^R, *e.g.*, is the reversal of the string a).

  FALSE. The correct right-hand side would be $ac^Rb^Re^Rd^R$, which in general is a different string from $ae^Rd^Rc^Rb^R$.

- (e) The **degree** of a node in an undirected graph is the number of its neighbors. If we add together the degrees of all the nodes in such an undirected graph, the result must be even.

  TRUE. Each edge contributes 2 to the sum of the degrees, so the total is twice the number of edges, which is an even number.

- (f) Recall that an undirected graph is defined to be **bipartite** if its nodes can be partitioned into two sets, such that every edge has an endpoint in each set. Then the undirected graph U from Question 4 is *not* bipartite.

  TRUE. There is a cycle of five nodes, from c to d to h to g to f and back to c. We can also see a back edge jumping four levels in the DFS tree, and an edge, (b, e), between adjacent levels in the BFS tree.

- (g) The operation $\rightarrow$ on boolean expressions is neither commutative nor associative.

  TRUE. The compound propositions $p \rightarrow q$ and $q \rightarrow p$ are not equivalent, and neither are $p \rightarrow (q \rightarrow r)$ and $(p \rightarrow q) \rightarrow r$.

- (h) Consider the boolean expression with infix representation "$(a \wedge b) \vee c$". Then this expression has prefix representation "$\vee \wedge abc$" and postfix representation "$abc \wedge \vee$".

  FALSE. The prefix representation is correct, but the correct postfix representation would be $ab \wedge c \vee$.

- (i) Let G be a directed graph with finitely many nodes. Suppose we conduct a DFS or a BFS in such a way that once a node has left the open list, it can never be put on the open list again. Then both searches are guaranteed to terminate.

  TRUE. However the open list is managed, if no node enters it more than once, it must be emptied after n moves, where n is the total number of nodes.

- (j) Let a and b be two naturals that are each less than or equal to the Fibonacci number $F(n)$. Then the Euclidean Algorithm, on inputs a and b, could not terminate with more than $n+3$ divisions.

  TRUE. We proved on the homework that the worst-case number is $n-2$, less than $n+3$.