Final

May 12, 2021

1. Consider the following pseudo-code of binary search and answer the following questions.

Algorithm 1 Binary Search algorithm

```
 0: function RecBinarySearch(low, high, A, x)
 1: if low = high and A[low] = x then
 2:     return low
 3: if low = high and A[low] ≠ x then
 4:     return -1
 5: mid ← ⌊(low + high)/2⌋
 6: if x ≤ A[mid] then
 7:     high ← mid
 8: if x > A[mid] then
 9:     low ← mid + 1
10: return RecBinarySearch(low, high, A, x)
```

(a) Suppose an input list $A = (2, 5, 8, 10, 13, 19, 21, 32, 37, 52)$ and you called RecBinarySearch(low=1, high=10, A, x=35). Give the values for variables low and high for each call to RecBinarySearch. Then give the final value return value. Note that the index of list starts with 1 in this problem, i.e. A[1] = 2, A[2] = 5 and so on. Your answer would look like this:

low = ?, high = ?

...

low = ?, high = ?

Return value: ?

(b) Use the recurrence formula to show that the worst case time complexity of RecBinarySearch is $\Theta(\log n)$, where n = high - low.

(c) A student added a debugging function f on the line 1 to print the elements in the range of [low, high] in the array (Algorithm 2). The runtime of f(low, high) is cn where c is a constant and n = high - low. What is the worst case time complexity of RecBinarySearch2? Prove your result using a recurrence relation.

Algorithm 2 Binary Search algorithm with a debug line

```
 0: function RecBinarySearch2(low, high, A, x)
 1: f(low, high)
 2: if low = high and A[low] = x then
 3:     return low
 4: if low = high and A[low] ≠ x then
 5:     return -1
 6: mid ← ⌊(low + high)/2⌋
 7: if x ≤ A[mid] then
 8:     high ← mid
 9: if x > A[mid] then
10:     low ← mid + 1
11: return RecBinarySearch2(low, high, A, x)
```

2. Directed Acyclic Graphs (DAGs) and Divisibility

(a) Draw the following DAG, $G = (V, E)$ where $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and $E = \{< u, v >: u$ divides v and v/u is prime $\}$.

(b) What edges must be added to the edges of the DAG G to create the reflexive, transitive closure of its edges?

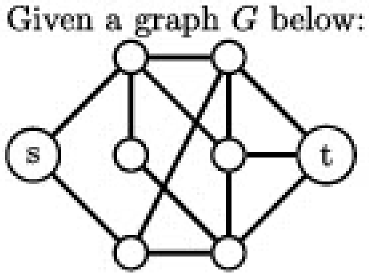
(c) Show that the divisibility relation $(p|q)$ is a partial order on $\mathcal{Z}^{>0}$.

USC CSCI 170 Final Exam

3. Suppose we have a simple undirected graph G with n vertices and every vertex in G has degree strictly more than $\lceil n/2 \rceil$. Prove that G is a connected graph.

4. Edge Connectivity

Given a graph G below:



(a) What is the edge connectivity of G?

(b) Given a simple graph, $G = (V, E)$, BFSCheckConnect will return true if the graph is connected and false otherwise.

Algorithm 3 BFSConnectCheck((V, E))

function BFSCONNECTCHECK((V, E))
Select the first vertex in V to be s
Set discovered$[s]$ = **true** and discovered$[v]$ = **false** for all other v
L[0] $\leftarrow \{s\}$
$i \leftarrow 0$
while L$[i]$ is not empty **do**
  Make L$[i+1]$ as empty list
  for all vertices $u \in$ L[i] **do**
    for all edges (u, v) **do**
      if discovered$[v]$ = **false then**
        discovered$[v] \leftarrow$ **true**
        Add v to list L$[i+1]$
  $i \leftarrow i + 1$
for all vertices $u \in V$ **do**
  if discovered$[u]$=**false then**
    return false
return true

Given a simple connected graph, $G = (V, E)$, ThreeConnectedCheck will return true if G is 3-connected and false otherwise.

Algorithm 4 ThreeConnectedCheck((V, E))

0: **function** THREECONNECTEDCHECK((V, E))
1: $T = \{\}$
2: **for all** vertices $u \in$ V **do**
3:   $T \leftarrow T \cup u$
4:   **for all** vertices $v \in V - T$ **do**
5:     $V^{'} \leftarrow V - \{u, v\}$
6:     $E^{'} \leftarrow E - \{e \in E \mid e \text{ is incident to } u \text{ or } v\}$
7:     **if** BFSConnectCheck($(V^{'}, E^{'})$) = **false then**
8:       **return** false
9: **return** true

State the loop invariant of the outer loop (starting on line 2) of ThreeConnectedCheck.

(c) Prove the correctness of ThreeConnectedCheck.

USC CSCI 170 Final Exam

5. The cost of the minimum spanning tree (MST) on complete graph

(a) Consider K_4. Now consider giving the edges of K_4 a weight. Each edge will have a distinct weight and the weight will be a value from 1 to 6. How can you assign the weights to the edges of K_4 such that the cost of the MST on K_4 will be the smallest possible? Explain your answer by drawing the graph with weights on edges and tracing the algorithm you used to construct the MST and why.

(b) Now consider the complete graph on n vertices with similar edge costs. More precisely, let $G = K_n$ with its edges given costs from 1 to $\frac{n(n-1)}{2}$. Show that for any $n \geq 2$ there exists an assigment of weights to the edges of G such that the cost of the MST on G is $\frac{n(n-1)}{2}$.

USC CSCI 170 Final Exam

6. Satisfiability and Bipartite graphs.

(a) Consider the expression $(p \oplus q) \wedge (\neg p \oplus q)$. A graph representing this expression will be the graph $G = \{V, E\}$ where $V = \{p, \neg p, q, \neg q\}$ and $E = \{(p, \neg p), (p, q), (\neg p, q), (q, \neg q)\}$. Is this expression satisfiable? If it is satisfiable, give a satisfying truth assignment. If it is not satisfiable, explain why it is not. Is its corresponding graph 2-colorable? If it is 2-colorable, give a valid 2-coloring for the graph. If it is not, justify why not.

(b) Consider the expression $(p \oplus q) \wedge (\neg p \oplus \neg q)$. A graph representing this expression will be the graph $G = \{V, E\}$ where $V = \{p, \neg p, q, \neg q\}$ and $E = \{(p, \neg p), (p, q), (\neg p, \neg q), (q, \neg q)\}$. Is this expression satisfiable? If it is satisfiable, give a satisfying truth assignment. If it is not satisfiable, explain why it is not. Is its corresponding graph 2-colorable? If it is 2-colorable, give a valid 2-coloring for the graph. If it is not, justify why not.

(c) Now more generally, consider a formula of propositional logic consisting of a conjunction of clauses of the form $(\pm p \oplus \pm q)$, where p and q are propositional variables (not necessarily distinct) and $\pm p$ stands for either p or $\neg p$. Consider the graph in which the vertices include p and $\neg p$ for all propositional variables p appearing in the formula, and in which there is an edge (a) connecting p and $\neg p$ for each variable p, and (b) connecting two literals if their exclusive-or is a clause of the formula. Prove that a formula is satisfiable if and only if its corresponding graph as described above is 2-colorable.