

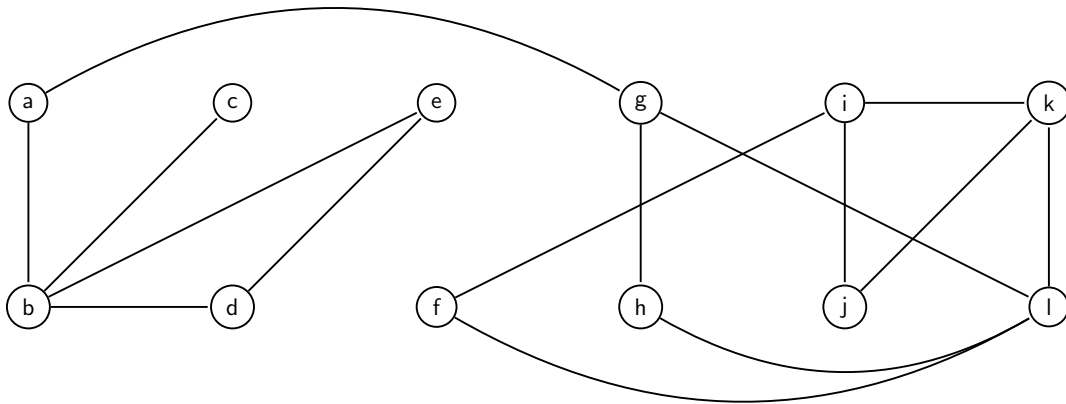
CS200 - Problem Set 8

1. Consider the generic graph search algorithm we discussed in class. Let s be the starting vertex. Prove that vertex v is visited by the algorithm if and only if there is a path from s to v in the graph.
2. In the NFL (National Football League), 10 players from each team are chosen each week to have a drug test. These choices are supposed to be random. Last year, a player named Eric Reid accused the league of over-testing him as retaliation for speaking out against the league, particularly for supporting players' right to kneel during the national anthem.

Assume there are 72 people on a team, 11 weeks in the season, 10 people are chosen from each team each week for a drug test, and each player is chosen with equal probability each week. Let E_i be the event that includes all elements of the sample space where a single player "Z" is chosen to be drug tested exactly i times over the course of the season. Determine $Pr(E_i)$ for each i from 0 to 5. Then calculate the probability that player Z has *at least* 6 tests. (You can use python or mathematica or wolfram alpha or another tool to help do these calculations. $\binom{n}{k}$ is often written as `binom[n,k]` or `nchoosek(n,k)` as a built-in function.)

Eric Reid was actually tested 6 times in 11 weeks. Please comment.

3. Let G be the following graph, and let A_G be an adjacency list representation of the G with the adjacency list for each vertex in alphabetical order.



Consider the following slight variation to breadth-first-search:

Algorithm 1: BFSish(A, s)

Input : Adjacency list A for a graph (V, E) and vertex s

Output: An integer array L of length $|V|$.

// Initialize array of explored vertices and array L

```
1  $X[v] = 0 \forall v \in V$ ;  
2  $L[v] = \infty \forall v \in V$ ;  
3  $X[s] = 1$ ;  
4  $L[s] = 0$ ;  
  // Initialize Queue  $Q$   
5  $Q = \{ \}$ ;  
6 Add  $s$  to  $Q$ ;  
7 while  $Q$  is not empty do  
8    $v = Q.pop$ ;  
9   for  $w \in A[v]$  do  
10    if  $X[w] = 0$  then  
11      $X[w] = 1$ ;  
12     Add  $w$  to  $Q$ ;  
13      $L[w] = L[v] + 1$ ;  
14    end  
15  end  
16 end  
17 return  $L$ 
```

- (a) In what order does $\text{BFSish}(A_G, a)$ explore the nodes of the graph G ? (Remember lists of the adjacency list representation are in alphabetical order, so the for loop in line 9 will look at vertices in alphabetical order.)
- (b) What are the values in the array L that is returned when $\text{BFSish}(A_G, a)$ is implemented?
- (c) Considering your answer to part b, what does it seem like the algorithm is doing? What is the meaning of L ? Think about the relationship between a , b , and $L[b]$.
4. (a) Let $T(n)$ be the number of bit strings of length n that have two consecutive zeros. (A bit string of length n is an element of $\{0, 1\}^n$.) Consider a recurrence relation for $T(n)$.
- What is the recurrence part of the recurrence relation?
 - What is the base case(s) for the recurrence relation?
 - Use the recurrence relation to calculate $T(5)$.
- (b) Consider the following variant to the Tower of Hanoi. We start with all disks on peg 1 and want to move them to peg 3, but we cannot move a disk directly between peg 1 and peg 3. Instead, we can only move disks from peg 1 to peg 2, or from peg 2 to peg 3. So in the case of $n = 1$ disk, we have to first move the disk from 1 to 2, and then from 2 to 3. Let $T(n)$ be the number of moves required to shift a stack of n disks from peg 1 to peg 3, if as usual, we can not put a larger disk on top of a smaller disk. Consider creating a recurrence relation for $T(n)$.
- What is the recurrence part of the recurrence relation? (Explain)

- ii. What is the base case for the recurrence relation?
 - iii. Use the iterative method to solve for $T(n)$ and give a big-O bound on $T(n)$.
5. We study the following algorithm in CS302.

Algorithm 2: `dynamic(n)`

Input: An $n \times 3$ array L containing natural numbers.
 $A, B \in \mathbb{N}$, a rectangular array Q of size $A \times B$ with all 0's

```

1 for k=1 to A do
2   for j=1 to B do
3     for q=1 to k-1 do
4       if  $Q[k, j] < Q[q, j] + Q[k - q, j]$  then
5         |  $Q[k, j] := Q[q, j] + Q[k - q, j]$ ;
6       end
7     end
8     for r=1 to j-1 do
9       if  $Q[k, j] < Q[k, r] + Q[k, j - r]$  then
10        |  $Q[k, j] := Q[k, r] + Q[k, j - r]$ ;
11      end
12    end
13    for i=1 to n do
14      if  $(k = L[i, 1] \text{ and } j = L[i, 2])$  or
15          $(k = L[i, 2] \text{ and } j = L[i, 1])$  then
16        | if  $Q[k, j] < L[i, 3]$  then
17          |  $Q[k, j] := L[i, 3]$ ;
18        end
19      end
20    end
21 end
22 return  $Q[A, B]$ ;

```

Last PSet, we found that an expression for the worst-case runtime is:

$$C_1 + \sum_{k=1}^A \left(\sum_{j=1}^B \left(\sum_{q=1}^{k-1} C_2 + \sum_{r=1}^{j-1} C_3 + \sum_{i=1}^n C_4 \right) \right) \quad (1)$$

- (a) Analyze the expression from part a by first evaluating all of the sums, and then put a big-O bound on your simplified expression.
 - (b) Explain how you can bound the asymptotic runtime without using summation notation, but instead using a rough worst-case analysis.
6. Suppose you have a weighted coin such that the probability of heads is 0.3 and the probability

of tails is 0.7. In this problem, you will calculate the probability of getting at least 3 heads, if you flip the coin 10 times.

- (a) First describe the tree (don't actually write it out!) that you can use to calculate the relevant probabilities. Use as many of the tree vocabulary words we learned in class as possible in your description.
 - (b) Next use the properties of this tree to calculate the probability of getting at least 3 heads, if you flip the coin 10 times.
7. Consider rolling 5 dice. Let $X_{i,j}$ be an indicator random variable that takes value 1 if the i th die has outcome j (and takes value 0 otherwise).
- (a) What is the sample space? What is its size?
 - (b) Let X be the random variable that is the sum of all of the values shown on the dice. If the outcome of the rolls is $s = (4, 2, 4, 5, 1)$, what is $X(s)$? What is $X_{3,4}(s)$? What is $X_{4,3}(s)$?
 - (c) Write X in terms of a *weighted* sum of the variables $X_{i,j}$.
 - (d) What is $\mathbb{E}[X_{i,j}]$?
 - (e) Use linearity of expectation to determine the average value of the sum of all values shown on the dice.
8. **[This problem is moved to the next problem set.]** Suppose a group of n people each order a different flavor of ice cream at an ice cream shop. Suppose the server didn't keep track of who ordered which flavor, and just handed the ice cream out randomly.
- (a) Let X a random variable that is the number of people who got handed the correct flavor. Let X_i be the indicator random variable that takes value 1 if person i gets the correct flavor (and 0 otherwise.) Write X in terms of a sum of the X_i .
 - (b) Use linearity of expectation to determine the average number of people who get the correct flavor.
9. How long did you spend on this homework?