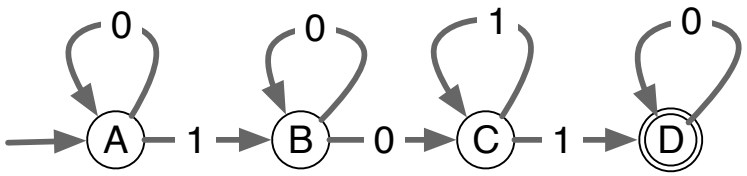
Midterm — Algorithms Fall 2020

Work on problems making up 80 points.

10 points (1) Give a regular expression that describes all strings over the alphabet $\{a, g, t\}$ that start with a and do not end with a.

10 points (2) The following NFA is to be converted to a DFA. What are the transitions corresponding to the new State {A,B}?



10 points (3) The following algorithm works on an $n \times n$ matrix. What is its run-time in terms of n?

```
def qu(mat):
   source = 0
   min_dist = [float('inf') for _ in range(n)]
   min_dist = 0
   for i in range(n-1):
      for v in range(n):
         for w in range(n):
            if min_dist[w]> min_dist[v] + mat[v][w]:
               min_dist[w] = min_dist[v] + mat[v][w]
   for v in range(n):
      for w in range(n):
         if min_dist[w]>min_dist[v]+mat[v][w]:
            return True
   return False
```

(4) HorridSort works like this to sort an array a with *n* elements. If $n < 9$, it uses Bubblesort. Otherwise, it calls itself six times. First, it calls itself on the slice of the array made up of the first half of the elements and sorts it. Then it calls itself on the slice of the middle n/2 elements. Then it calls itself on the slice of the last half of the elements. Then it repeats these calls.

10 points (a) Give a recursion formula (with the conventions of the master theorem) for HorridSort.

10 points (b) Use the master theorem to solve this recursion and to determine the asymptotic run-time of HorridSort.

70 points (c) [Not advised] Prove or disprove the correctness of HorridSort.

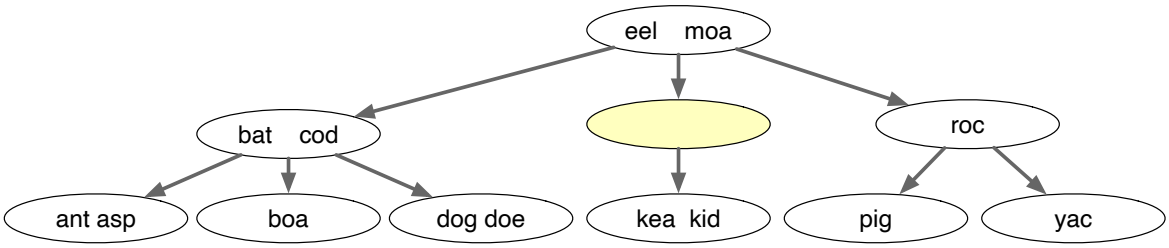
(5) Simultaneous min-max pairs up the elements of an array and then finds the maximum among the larger of the pairs and the minimum among the smaller of the pair. Evaluate instead the following algorithm: We divide the elements in the array into groups of three.

(One group might have less than three elements). We use three comparisons in order to find the maximum and the minimum for each group of three. We then use $m - 1$ comparisons to find the minimum of the m minima of the groups of threes and the same number of comparisons for the maximum of the m maxima of the groups of three.

10 points (a) If n is a multiple of three, how many comparisons are made. Do we save over the naïve algorithm with $2(n - 1)$ comparisons?

10 points (b) If n has remainder one after division with three, how many comparisons do we make? How do we compare with $2(n - 1)$ comparisons for the naïve algorithm?

10 points (6) During deletion, a B-tree has the following form. The yellow node suffers an underflow.



Describe the next operation (or draw the result). In particular, which nodes will be children of the yellow node.

10 points (7) Give the level and the split pointer for an LH structure with 100 buckets. Where would the record with hash of key 100 go?

20 points (8) What follows is the classical maximum of an array of numbers solution.

```
def max(A):
   answer = A[0]
   for j in range(1, len(A)):
      if A[j] > answer:
         answer = A[j]
   return answer
```

Show by induction that the loop invariant: `answer is the maximum of A[0:j+1]` is maintained through each iteration of the loop.