CS 412: Algorithms: Design and Analysis, L4, Quiz 2B, Spring 2023.

14 problems for 60 points on 5 printed sides. Duration: 75 minutes. 1300-1415h, March 13, 2023.

Instructions:

1. Enter your name and ID below and at the top of every side.

Student Name: ______________________________

Student ID: ______________________________

Mutliple Choice

For each of the following multiple-choice questions, choose *one* of the five choices (A, B, C, D or E) and indicate it in the answer grid provided at the end of this section. The answer to Question 0 is already entered for you in the grid.

0. (0 points) This is a sample question. The number of characters in the word “Algorithm” are:

   A. 8 **B. 9** C. 10 D. 11 E. 12

1. (2 points) Which of the following sets is $f(n) = n^3 + 2^{n+1}$ NOT a member of?

   A. $O(2^n)$ **B.** $o(n^3)$ C. $\Theta(2^n)$ D. $\omega(n^2)$ E. $\Omega(2^n)$

2. (2 points) Let $f(n) \le g(n)$ denote $f(n) = O(g(n))$. Indicate which of the following is correct:

   A. $\lg\lg n \le \lg n \le \lg^2 n$ B. $\lg n \le \lg\lg n \le \lg^2 n$ C. $lgn \le \lg^2 n \le \lg\lg n$
   D. $\lg^2 n \le \lg\lg n \le \lg n$ E. $\lg\lg n \le \lg^2 n \le \lg n$

Questions 3 and 4 refer to the to the following code snippet that computes x^n recursively.

```
def power(x: int, n: int):
  if n == 1:
    return n
  if n % 2 == 0:
    return power(x, n//2) * power(x, n//2)
  return x * power(x, n-1)
```

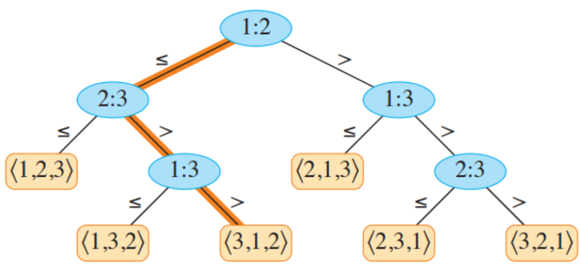
3. (2 points) What is the recurrence for $T(n)$, the time taken to compute `power(x, n)` in the best case, i.e. when $n = 2^k$?

   A. $T(n) = T\left(\frac{n}{2}\right) + \Theta(1)$ B. $T(n) = T(n-1) + \Theta(1)$ **C.** $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(1)$ D. $T(n) = 2T(n-1) + \Theta(1)$ E. $T(n) = T\left(\frac{n}{2}\right) + T(n-1) + \Theta(1)$

4. (2 points) What is the asymptotic complexity for $T(n)$ for the best case?

   A. $T(n) = \Theta(1)$ B. $T(n) = \Theta(\lg n)$ **C.** $T(n) = \Theta(n)$ D. $T(n) = \Theta(n \lg n)$ E. $T(n) = \Theta(n^2)$

Questions 5 and 6 refer to the to the following description.



Recall from the section on “A Lower Bound on Sorting” that we introduced the concept of a “Decision Tree”. A decision tree is a binary tree that is composed of nodes that represent a comparison of two elements. For example, the tree on the left shows a sorting of 3 elements by comparing the elements at indices 1, 2 and 3. This tree has 5 internal nodes representing 5 comparisons.

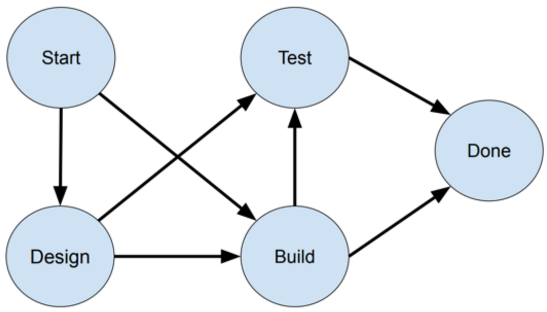
5. (2 points) How many internal nodes are there in a decision tree for sorting 4 elements?

   A. 5 B. 11 C. 17 **D. 23** E. 29

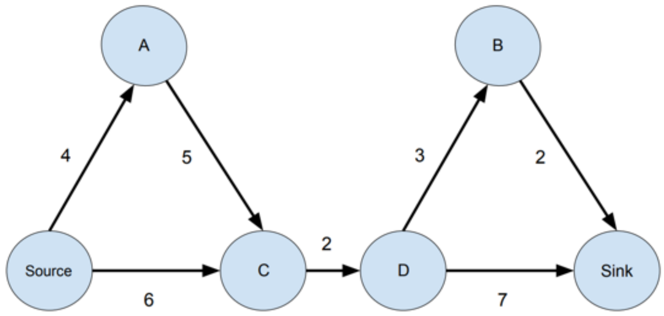
6. (2 points) In general, for sorting n elements, the number of internal nodes (i.e. the number of times we compare elements) is:

   A. $n!$ B. $\frac{n!}{2}$ C. $(2n)!$ **D.** $n! - 1$ E. $n! + 1$

Questions 7 to 10 refer to the following graphs.



(a)



(b)

7. (2 points) Which of the following order of vertices is a valid topological sort of the graph shown in (a)?

   A. Start, Design, Build, Test, Done

   B. Start, Design, Test, Build, Done

   C. Start, Build, Design, Test, Done

   D. Start, Test, Design, Build, Done

   E. Start, Build, Test, Design, Done

8. (2 points) Which of the following edges, if added to the graph shown in (a), will result in a graph for which topological sort is not possible?

   A. `Start` → `Test` B. `Design` → `Done` **C.** `Test` → `Design` D. `Test` → `Done` E. `Start` → `Done`

9. (2 points) What is the flow across a min-cut of the flow network shown in (b), i.e., the minimum capacity across all cuts?

   A. 2 B. 3 C. 4 D. 5 E. 6

10. (2 points) What is the maximum flow across the flow network shown in (b) if an edge with capacity 10 is added vertex A to vertex B?

    A. 2 B. 3 **C. 4** D. 6 E. 12

Q0. **B**	Q1. **B**	Q2. **A**
Q3. **C**	Q4. **C**	Q5. **D**
Q6. **D**	Q7. **A**	Q8. **C**
Q9. **A**	Q10. **C**	Total

Long Answers

Attempt the following problems in the provided answer sheet. Use the back of the sheet for rough work, if required.

1. (10 points) Algorithm A operates on a sequence of size n by recursively computing the solution for the left and right halves of the sequence, and then performing a total of $2n+1$ steps on the obtained results. Algorithm B computes the same result by performing 7 steps on the result obtained on a subsequence of size $(n-1)$. Both algorithms perform 3 steps on a sequence whose size is 1.

   Given a sequence of size $2^{16} = 65536$, which algorithm will compute the result in fewer steps? Justify your answer.

Solution: The recurrence for the number of steps in each case is:

$$T_A(n) = 2T_A\left(\frac{n}{2}\right) + 2n + 1,\ T_B(n) = T_B(n-1) + 7,\ T_A(1) = T_B(1) = 3$$

Solving for T_A,

$$\begin{aligned}
T_A(n) &= 2T_A\left(\frac{n}{2}\right) + 2n + 1 \\
&= 2\left(2T_A\left(\frac{n}{4}\right) + n + 1\right) + 2n + 1 = 4T_A\left(\frac{n}{4}\right) + 4n + 3 \\
&= 4\left(2T_A\left(\frac{n}{8}\right) + \frac{n}{2} + 1\right) + 4n + 3 = 8T_A\left(\frac{n}{8}\right) + 6n + 7 \\
&= 8\left(2T_A\left(\frac{n}{16}\right) + \frac{n}{4} + 1\right) + 6n + 7 = 16T_A\left(\frac{n}{16}\right) + 8n + 15 \\
&= 2^i T_A\left(\frac{n}{2^i}\right) + 2in + 2^i - 1
\end{aligned}$$

Putting $2^i = n$,

$$T_A(n) = nT_A(1) + 2n\lg n + n - 1, \qquad \boxed{\therefore T_A(n) = 2n\lg n + 4n - 1}$$

Solving for T_B,

$$\begin{aligned}
T_B(n) &= T_B(n-1) + 7 \\
&= T_B(n-2) + 14 \\
&= T_B(n-3) + 21 \\
&= T_B(n-i) + 7i
\end{aligned}$$

Putting $n - i = 1$,

$$\begin{aligned}
T_B(n) &= T_B(1) + 7(n-1) \\
&= 3 + 7n - 7
\end{aligned}$$

$$\boxed{\therefore T_B(n) = 7n - 4}$$

For $n = 2^{16}$,

$$\begin{aligned}
T_A(n) &= 2 \cdot 2^{16} \lg 2^{16} + 4 \cdot 2^{16} - 1 \\
&= 32 \cdot 2^{16} + 4 \cdot 2^{16} - 1 \\
&= 36 \cdot 2^{16} - 1 \\
&= 9 \cdot 2^{18} - 1
\end{aligned}
\qquad
\begin{aligned}
T_B(n) &= 7 \cdot 2^{16} - 4 \\
&=
\end{aligned}$$

The quantity for T_B is smaller. Thus, Algorithm B will take fewer steps.

2. (15 points) Given a sorted sequence, A, of n distinct integers, $\langle A_1, A_2, A_3, \ldots, A_n \rangle$, you want to find out whether there is an index i for which $A_i = i$.

   (a) Provide an efficient algorithm for this task.

   Solution:

```
# index ranges are inclusive of endpoints
def element_equals_index(A, start, end):
    if start == end:  # base case - single element
        return A[start] == start
    if start > end:  # base case - empty sequence
        return False
    mid = (start + end) // 2
    if A[mid] == mid:  # found at mid
        return True
    elif A[mid] < mid:  # possibility in upper half
        return element_equals_index(A, mid+1, end)
    else:  # possibility in lower half
        return element_equals_index(A, start, mid-1)
```

   (b) Argue why your algorithm is correct.

   Solution: The algorithm is very similar to binary search, it either finds the answer at the middle elements or discards half the search space for the next recursive call.

   The key observation is as follows. Because the sequence is sorted and contains distinct elements, if the middle element is smaller then its index, then all the preceding elements are be smaller than their index. If the sequence contains an element that is equal to its index, it must lie in the half following the middle.

   Similarly, if the middle element is larger than its index, then all the following elements are also larger than their index. If the sequence contains an element that is equal to its index, it must lie in the half preceding the middle.

   (c) Argue about the time complexity of your algorithm.

   Solution: The algorithm operates very similarly to binary search and therefore has a time complexity of $O(\lg n)$.

   You get up to 5 points if you do the above tasks correctly and your algorithm takes $O(n)$ time. You may get more marks if your algorithm is more efficient.

3. You are organizing students into teams for the upcoming HUSG election. So that each team has a uniform philosophy, only students who are influenced by each other are put in the same team. There is no restriction on the total number of teams but a team may have no more than 4 students.

   A student A is influenced by another student B if A follows B on social media or follows someone who is influenced by B.

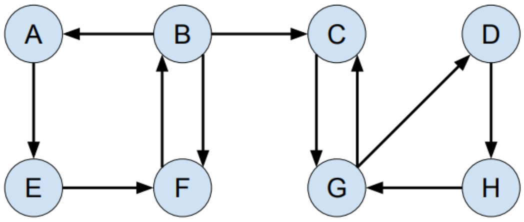
   8 students are interested in running. For convenience, we have labeled them as A, B, C, D, E, F, G, and H. From their social media profiles, we have gathered the following data:

   - A follows E;
   - B follows A, C and F;
   - C follows G;
   - D follows H;
   - E follows F;
   - F follows B;
   - G follows C and D;

- H follows G.

(a) (5 points) Describe and show an appropriate representation of the information of the 8 students.

Solution:



The information can be represented as a directed graph, $G = (V, E)$, where $V = \{A, B, C, D, E, F, G, H\}$ and $E = \{(u, v) \mid u, v \in V, u \text{ follows } v\}$.

(b) (5 points) Describe an algorithm that can be used to identify teams from such information.

Solution: The problem can be formulated as finding the strongly connected components which correspond to teams of students that are influenced by each other.

The algorithm to find strongly connected components algorithm (SCC) requires two DFS traversals, one through G and another through G^T, its transpose.

(c) (5 points) Apply the algorithm to the 8 students above and mention the steps taken in order to identify the teams. State the teams obtained.

Solution: Applying the SCC Algorithm yields two teams of 4 each: A, B, E, F and C, D, G, H.