There are two assignments in this handout. Do all but submit the following.

1. **Assignment 4**: Problems 3, 5, 6, 9, 11 by **Tuesday April 21 in class**
2. **Assignment 5**: Problems 12, 13, 14, 15, 17 by **Tuesday April 28 in class**

**Announcement:**
- Last lecture May 5
- Final exam 4:30am - 7 pm Saturday May 9 (please verify)
- For the final exam, focus on materials beyond midterm. Getting grade B and above require understanding course materials beyond the course assignments.
- Chapters covered in this class (text 3rd edition): Foundation (1-5), Sorting and Order statistics (6-9), Advanced design and algorithms (15-17), Graph Algorithms (22-23, part of 24 if time allowed), NP-Completeness (34).

1. Why do we want the loop index \( i \) in line 2 of BUILD-MAX-HEAP as shown below to decrease from \( \lceil \text{length}[A]/2 \rceil \) to 2 rather than increase from 1 to \( \lceil \text{length}[A]/2 \rceil \)?

   ```
   BUILD-MAX-HEAP(A)
   1 heap-size[A] ← length[A]
   2 for i ← ⌈length[A]/2⌉ downto 1
   3 do MAX-HEAPIFY(A, i)
   ```

2. Using the Figures below as a model, illustrate the operation of HEAPSORT on the array \( A = \langle 5, 13, 2, 25, 7, 17, 20, 8, 4 \rangle \).

   ![Diagram](image)

3. What is the running time of heapsort on an array \( A \) of length \( n \) that is already sorted in increasing order? What about decreasing order?
4. Using the Figures below as a model, illustrate the operation of \textsc{Partition} on the array \( A = \langle 13, 19, 9, 5, 12, 8, 7, 4, 11, 2, 6, 21 \rangle \).

5. What value of \( q \) does \textsc{Partition} return when all elements in the array \( A[p..r] \) have the same value?

6. What is the running time of \textsc{QuickSort} when all elements of array \( A \) have the same value?

7. Banks often record transactions on an account in order of the times of the transactions, but many people like to receive their bank statements with checks listed in order by check number. People usually write checks in order by check number, and merchants usually cash them with reasonable dispatch. The problem of converting time-of-transaction ordering to check-number ordering is therefore the problem of sorting almost-sorted input. Argue that the procedure \textsc{Insertion-Sort} would tend to beat the procedure \textsc{QuickSort} on this problem.

8. Why do we analyze the average-case performance of a randomized algorithm and not its worst-case performance?

9. Illustrate the operation of \textsc{Counting-Sort} on the array \( A = \langle 6, 0, 2, 0, 1, 3, 4, 6, 1, 3, 2 \rangle \).

10. Suppose that the \textbf{for} loop header in line 9 in the Figure below of the \textsc{Counting-Sort} procedure is rewritten as “\textbf{for} \( j \leftarrow 1 \) to \( \text{length}[A] \)” Show that the algorithm still works properly. Is the modified algorithm stable?

\begin{verbatim}
COUNTING-SORT(A, B, k)
1     for i ← 0 to k
2         do C[i] ← 0
3     for j ← 1 to length[A]
4         do C[A[j]] ← C[A[j]] + 1
5     \( \triangleright \) C[i] now contains the number of elements equal to \( i \).
6     for i ← 1 to k
7         do C[i] ← C[i] + C[i - 1]
8     \( \triangleright \) C[i] now contains the number of elements less than or equal to \( i \).
9     for j ← length[A] downto 1
11     C[A[j]] ← C[A[j]] - 1
\end{verbatim}
11. Using the Figure below as a model, illustrate the operation of RADI-SORT on the following list of English words: COW, DOG, SEA, RUG, ROW, MOB, BOX, TAB, BAR, EAR, TAR, DIG, BIG, TEA, NOW, FOX.

12. Show how to sort \( n \) integers in the range 0 to \( n^3 - 1 \) in \( O(n) \) time.

13. Using the Figures below as a model, illustrate the operation of BUCKET-SORT on the array \( A = \langle .79, .13, .16, .64, .39, .20, .89, .53, .71, .42 \rangle \).

14. What is the worst-case running time for the bucket-sort algorithm? What simple change to the bucket-sort algorithm preserves its linear expected running time and makes its worst-case running time \( O(n \lg n) \)?

15. Find an optimal parenthesization of a matrix-chain product whose sequence dimension is \( \langle 5, 10, 3, 12, 5, 50, 6 \rangle \).

16. Which is a more efficient way to determine the optimal number of multiplications in a chain-matrix multiplication problem: enumeration all possibilities or running a recursive-matrix-chain? Justify your answer.

17. Explain why memorization is ineffective in speeding up a good divide-and-conquer algorithm such as merge-sort?