

“Homework problem session” is in CSB 209, 7:30-8:30pm on Dec. 3 (Wednesday); held by Sean Esterkin.

Homework is due Tuesday, Dec. 4 before class.

“Exam problem session” is in CSB 209, 6:00pm-7:00pm on Dec. 10 (Wednesday); held by the instructor.

EXAM #4 will be on **Thursday, Dec. 11**.

1 Homework - solve and turn in

5.1 (due Thursday, Dec 4) Let A and B be two sets. Their sum $A + B$ is defined to be

$$A + B := \{a + b \mid a \in A, b \in B\},$$

for example if $A = \{1, 4, 6\}$ and $B = \{2, 5\}$ then $A + B = \{3, 6, 8, 9, 11\}$. Give $O(n \log n)$ algorithm whose input is two sets $A, B \subseteq \{1, \dots, n\}$ and the output is set $A + B$.

5.2 (due Thursday, Dec 4) The Hadamard matrices H_0, H_1, \dots , are defined as follows.

- H_0 is the 1×1 matrix $[1]$.
- For $k > 0$, H_k is the $2^k \times 2^k$ matrix

$$H_k = \left[\begin{array}{c|c} H_{k-1} & H_{k-1} \\ \hline H_{k-1} & -H_{k-1} \end{array} \right].$$

Show that if v is a column vector of length $n = 2^k$ then the matrix-vector product $H_k v$ can be calculated using $O(n \log n)$ operations. (Assume that all the numbers involved are small enough that basic arithmetic operations take $O(1)$ time.)

5.3 (due Thursday, Dec 4) We are given k sorted lists A_1, \dots, A_k . The total number of elements in the lists is n . We would like to merge the lists into one sorted list B (the number of elements in B will be n). Give an algorithm which solves this problem in time $O(n \log k)$.

5.4 (due Thursday, Dec 4) We are given an array of integers $A[1..n]$ which is almost sorted in the following sense: for all $i \in \{1, \dots, n - k\}$ we have $A[i] \leq A[i + k]$. Give an algorithm which sorts the array A . Your algorithm should run in time $O(n \log k)$.

5.5 (due Thursday, Dec 4) We are given an array of integers $A[0..n - 1]$. Dr. Median uses the following program

```
for i from 0 to n - 1 do
  for j from 0 to n - 1 do
    C[i+n*j] = A[i]+A[j];
return the median of C[0], C[1], ..., C[n^2 - 1].
```

Dr. Median needs a faster program for the task. Give an $O(n \log n)$ algorithm that returns the same answer as the algorithm above.

2 Bonus Homework - solve and turn in

In problems 5.6, 5.7, 5.8, **do NOT use hashing** (in practice hashing would be a great technique for these problems even though it does not have worst-case guarantees), **do NOT use linear sorting** (the assumptions of the problems do not allow for a linear sorting algorithm, such as, radix-sort).

5.6 (due Thursday, Dec 4) We are given an array A with n elements and a number C . Assume that the sum of the elements in A is larger than C . We would like to compute the size of the smallest subset of A whose elements sum to at least C . (For example, if $A = [8, 3, 9, 2, 7, 1, 5]$ and $C = 18$ then the answer is 3; the set is $\{7, 8, 9\}$.) Give a linear-time algorithm for this problem. (HINT: use the linear-time SELECT algorithm)

5.7 (due Thursday, Dec 4) We are given an array of integers $A[1..n]$. We would like to determine whether there exists an integer x which occurs in A more than $n/2$ times (i. e., whether A has a majority element). Give an algorithm which runs in time $O(n)$. (HINT: use the linear-time SELECT algorithm.)

Example: For $A = [3, 1, 2]$ the answer is NO. For $A = [3, 1, 3]$ the answer is YES.

5.8 (due Thursday, Dec 4) We are given an array of integers $A[1..n]$. We would like to determine whether there exists an integer x which occurs in A more than $n/3$ times. Give an algorithm which runs in time $O(n)$. (HINT: use the linear-time SELECT algorithm)

5.9 (due Thursday, Dec 4) We are given two sorted arrays A and B , each containing n elements. Assume that the arrays do not contain duplicates, and the elements in A are different from elements in B . We would like to compute the median of $A \cup B$. For example if $A = [1, 2, 3, 4, 5]$ and $B = [6, 7, 8, 9, 10]$ then the median is 5; if $A = [1, 3, 5, 7, 9]$ and $B = [2, 4, 6, 8, 10]$ then the median is again 5. Give an $O(\log n)$ algorithm for this problem.

3 Additional problems from the book (do not turn in)

Try to solve the following problems. A few of them may be on the exam. We will go over the ones that you choose in the problem sessions.

- 2.4, 2.5, 2.8, 2.10, 2.12, 2.16, 2.17, 2.18, 2.21, 2.26, 2.27, 2.28, 2.29, 2.30, 2.31, 2.34.