Problem 1.

1. The "bottom" of the recursion is when $n = 2$, in which case the problem is solved with a single comparison. For $n > 2$, recursively solve the problem on half of the elements (say this returns a_1 as the maximum and b_1 as the minimum), then recursively solve it on the other half (say this returns a_2 and b_2), and finally compute and return the overall maximum as the larger of a_1 and a_2, and the overall minimum as the smaller of b_1 and b_2.

2. $T(2) = 1$. If $n > 2$ then $T(n) = 2T(n/2) + 2$.

3. (a) $T(n) = 2T(n/2)+2 = 2(2T(n/2^2)+2)+2 = \ldots = 2^iT(n/2^i)+2^i+2^{i-1}+\cdots+2$
   If, in the above, we let S_i denote $2^i + 2^{i-1} + \cdots + 2$, then
   $2S_i = 2^{i+1} + 2^i + \cdots + 2^2 = 2^{i+1} + S_i - 2$.
   Therefore $S_i = 2^{i+1} - 2$ and the recurrence becomes
   $T(n) = 2^iT(n/2^i) + 2^{i+1} - 2$.
   When $n/2^i$ equals 2, the above becomes
   $T(n) = (n/2)T(2) + n - 2 = (3n/2) - 2$.

   (b) *Basis of induction.* For $n = 2$ the right-hand-side is $(3 * 2/2) - 2 = 1$, which is $T(1)$ as required.
   Inductive step. Inductive hypothesis: Assume the claim holds up to (and including) n. To show that it must also hold for $2n$, note that the recurrence relation and inductive hypothesis imply the following:
   $T(2n) = 2T(n) + 2 = 2((3n/2) - 2) + 2 = 3n - 4 + 2 = 3n - 2 = 3(2n)/2 - 2$.

   Note that the answer is the same as for the non-recursive algorithm.

Question 2. Select the kth smallest item (call it x) in linear time using the selection algorithm covered in class, then go through S and create the set $S_{\leq x}$ that contains the k elements of S that are $\leq x$. Sort $S_{\leq x}$. Everything takes linear time except the sorting that takes $O(k \log k)$, hence the total time is $O(n + k \log k)$.

Question 3. Choosing y to be the median of the set $\{y_1, \ldots, y_n\}$ minimizes the quantity $f(y) = \sum_{i=1}^{n} |y - y_i|$. Finding such a y can be done in $O(n)$ time by using the linear time selection algorithm (that algorithm was covered in class). Correctness follows from the following:

1. The function $f(y)$ we seek to minimize is piecewise linear, convex, and consists of $n+1$ straight-line pieces whose slopes are (in left-to-right order) $-n, -n+2, \ldots, n-2, n$.

2. If n is even then the minimum of $f(y)$ occurs on a straight-line piece of zero slope: Any point on that piece minimizes $f(y)$, including the left endpoint of that piece, which happens to correspond to the median of $\{y_1, \ldots, y_n\}$.

3. If n is odd then there is no straight-line piece of zero slope, but there is a straight-line piece of slope -1 followed by a straigh-line piece of slope $+1$: The minimum of $f(y)$ occurs on the point common to these two pieces, which happens to correspond to the median of $\{y_1, \ldots, y_n\}$.

Question 4. We first show that it suffices to compute, in $O(n^2)$ time, an $n \times n$ matrix S where $S[i, j]$ is the summation

$$\sum_{0 \le a \le i, 0 \le b \le j} A[a, b]$$

This is because, with the help of the matrix S, we can compute each r_i in constant time:

$$r_i = S[i_2, j_2] - S[i_2, j_1 - 1] - S[i_1 - 1, j_2] + S[i_1 - 1, j_1 - 1]$$

with the notational convention that $S[a, b]$ is zero if $a = -1$ or $b = -1$. To see that the above equation for r_i is correct, draw a picture of region R_i and notice that $S[i_2, j_2]$ corresponds to a region that contains R_i but also contains 3 other unwelcome rectangles whose sums need to be subtracted out; but subtracting both $S[i_2, j_1 - 1]$ and $S[i_1 - 1, j_2]$ from $S[i_2, j_2]$ is an over-correction because one of the 3 unwelcome rectanges has now been subtracted out twice, which is why we fix the over-correction by adding it back once (by adding $S[i_1 - 1, j_1 - 1]$).

We now show how the S matrix can be computed in $O(n^2)$ time. We do so by first computing a "horizontal" summation matrix (denoted H) where $H[i, j]$ is

$$\sum_{0 \le k \le j} A[i, k]$$

Computing H is easy to do in $O(n^2)$ time by computing each of its rows in $O(n)$ time with a left-to-right walk in A along that row. Once we have H, we can obtain S in $O(n^2)$ time as follows (where we use the notational convention that $S[i, j]$ is zero if $i = -1$ or $j = -1$):

- For $i = 1, \ldots, n$ in turn, compute row i of S as follows:
  - For $j = 0, \ldots, n - 1$ set $S[i, j]$ equal to $H[i, j] + S[i - 1, j]$